On the Achievable Forwarding Capacity of an Infinite Wireless Network

Jarno Nousiainen
jarno.nousiainen@tkk.fi

Jorma Virtamo
jorma.virtamo@tkk.fi

Pasi Lassila
pasi.lassila@tkk.fi

Aalto University, Department of Communications and Networking
P.O.Box 13000, FI-00076 Aalto, Finland

ABSTRACT

We consider the problem of finding the maximum directed packet flow that can be sustained in an infinite wireless multihop network. This ability of the network to relay traffic is called the forwarding capacity, and the problem appears when the spatial scales corresponding to the end-to-end paths (routing) and the neighboring nodes (forwarding) are strongly separated in a massively dense network. We assume a Boolean interference model. The infinite network is approximated with a finite but large network where the node locations form a spatial Poisson process. We study two constructive approaches to tighten the lower bound for the forwarding capacity by a significant amount. In *path scheduling* the packets traverse the network using predefined paths that do not interfere with each other, and coordination is thus required only between the nodes of a path. In *greedy maximum weight scheduling*, the transmissions are scheduled greedily according to queue-length based weights of the links. In addition to a fixed transmission radius, we consider greedy maximum weight scheduling with a transmission radius adjustable up to a given maximum. We are able to produce numerical results that characterize the achievable forwarding capacity under global coordination of the transmissions, providing, e.g., concrete points of reference for practical distributed implementations.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: Modeling techniques; C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless communication*

General Terms

Algorithms, Performance, Design

Keywords

Wireless Multihop Networks, Forwarding Capacity, Flow Networks

1. INTRODUCTION

Analyzing the fundamental capacity limits of large multihop wireless networks is difficult due to the complex interaction between spatially distributed nodes, interference caused by simultaneous transmissions and the achievable transmission rates. Important insights on the asymptotic scaling of the capacity have been obtained in the seminal paper by Gupta and Kumar [9] and more recently by Franceschetti *et al.* [8]. However, the scaling analysis still leaves the constant of proportionality undefined. Our aim is to provide numerical results that fix the constant of proportionality in a particular setting referred to as dense networks.

In a dense network, nodes with very short transmission ranges practically cover the entire domain of the network, and a typical path between a source and a destination consists of a large number of hops. From a single node's point of view, this kind of a network appears to be extremely large or even infinite, and the amount of relay traffic traveling through the node is much higher than the amount of originating or terminating traffic.

Specifically at the limit of a so-called massively dense network, there is a strong separation between the microscopic level of neighboring nodes and the macroscopic level of end-to-end connections. Hence, the network capacity problem separates into two independent problems, see [10, 11]. At the macroscopic level, the underlying network is considered a continuous medium where the routes are smooth geometric curves (this continuum approach has been adopted, e.g., in [4, 12, 14]). The problem at the macroscopic level is to distribute the load equally throughout the network. This load balancing problem has been analyzed in [10, 11].

At the microscopic level, where the interest of this paper lies, the network appears to be an infinite random geometric network where the nodes are distributed according to a spatial Poisson process, and interference is modeled using the so-called Boolean interference model. We operate in a parameter region where most of the nodes are connected, i.e., above the percolation threshold [6], and form what we call a transport network, somewhat similar to the construction of [8]. Assuming that the traffic flowing in different directions is handled via time sharing, the specific task is to maximize the achievable forwarding capacity of the network in a given direction by optimal coordination of transmissions. The forwarding capacity of the system is expressed as the mean density of progress, see [1, 22].

In this paper, we study constructive ways to approach the maximum achievable forwarding capacity that complement the earlier upper bounds [18, 19, 20]. The infinite network is approximated by a large finite network where the relay traffic is generated by adding a strip of sources and sinks to opposite edges of the network that resides in a unit square. We use two approaches to arrive at lower bounds for the forwarding capacity. In the first one, we construct an algorithm that searches a densely packed set of pairwise non-interfering paths. Within these paths, scheduling of the links is trivial, as under the used interference model and fixed transmission radius, every third link can be active simultaneously. The path algorithm is computationally efficient and can be applied to very large network realizations.

In the second approach, we apply maximum weight scheduling [24] where we simulate a large network in discrete time and the decision about the resource allocation in each time slot is based on the current queue-length-based weight of each link. It is well known that maximum weight scheduling is able to achieve the capacity limit. This means that if it were possible to realize the scheme ideally, the simulations would result in the actual value for the sought-after forwarding capacity. However, the ideal application of maximum weight scheduling entails solving the maximum weight independent set problem in every time slot, which is infeasible for the network sizes we consider. Hence, we use a greedy algorithm which can be realized more efficently but, being suboptimal, results in a lower bound for the forwarding capacity. The approach is covered both in the case where the transmission radius is fixed and in the case where it can be reduced (by power control) so as to just reach the destination to minimize the interference.

Under our network assumptions, the forwarding capacity depends only on the mean node degree. The achieved forwarding capacity is evaluated as a function of this, and also of the network size to allow extrapolation of the results to an infinite network. After extensive numerical simulations with networks up to 1000's of nodes and node degrees up to 25, we have been able to determine the forwarding capacity as a function of the node degree with reasonable accuracy, even taking into account our greedy algorithm for solving the independent set. As far as we are aware of, our results represent the currently most accurate available numerical values for the achievable forwarding capacity under global coordination. Notably, the earlier lower bound obtained by using the locally coordinated ExOR protocol [2] is improved by factor 2. The results provide an important point of reference against which more practical distributed solutions can be compared.

The remainder of this paper is structured as follows. Section 2 presents related work. In Section 3, we present the used notation and the network model. Section 4 ties the forwarding capacity problem to a wider scope of dense multihop networks. Sections 5 and 6 present the path scheduling and the greedy maximum weight scheduling algorithms, while the numerical results are given in Section 7. Finally, Section 8 concludes the study.

2. RELATED WORK

The literature on wireless multihop network capacity modeling and analysis is vast and here we only touch on the most relevant parts from the point of view of this paper. It is well-known that for a given finite network the capacity maximization can be expressed as a linear programming (LP) problem. The inherent computational complexity of the LP problem can be eased by using only a subset of the necessary constraints resulting in upper/lower bounds [13, 15]. We have applied this in the context of the forwarding capacity in [18]. An upper bound can also be obtained by considering the maximum achievable forwarding capacity in a *single time slot*. This spatial reuse problem has been considered in [19, 20].

Our main results are related to the application of the max weight scheduling principle. It was introduced by Tassiulas and Ephremides in [24] where they present a centralized, dynamic scheduling and routing algorithm that is optimal in the sense that it will stabilize the network for any arrival rate vector whenever possible. Since its introduction, it has been shown to stabilize the system under more general settings with time varying rates [17]. While the principle behind max weight is very elegant, the practical limitation is the need to solve the maximum weight independent set problem. This has lead researchers to analyze the impact of imperfect scheduling, like greedy ones, see [5, 16, 25]. The results have shown that greedy approaches typically can achieve a fraction of the maximal stability region, but the fraction is difficult to characterize and heavily depends on the assumptions, e.g., about the interference model. It is also known that randomized maximum weight algorithms stabilize the system provided that there is a strictly positive probability that the true maximum weight independent set is selected, at least when interference is modeled using graph theoretic models, see [7, 23].

In this paper, we have applied a simple greedy approach for the scheduling. However, we believe that the loss in the efficiency of the scheduling due to our greedy approach in the random network setting and simple traffic scenario (traffic in a single direction) may not be that great. On the other hand, applying the randomized versions would plausibly make the dynamic behavior of the scheduling within a time slot much more random, which would, in turn, entail significantly longer simulation times in order to credibly sample the steady state characteristics of the queue processes of the simulated network. Thus, in view of the size of the networks we are aiming at in our study, the greedy method provides a practical approach and has allowed us to numerically credibly verify that the network flow has reached steady state.

3. PRELIMINARIES

In this section, we present the network model and the definitions used in the microscopic level model.

3.1 Network model

The network consists of nodes distributed randomly over a plane according to a spatial Poisson point process with density λ. We operate with slotted time, and the length of a time slot matches the length of a packet transmitted with the nominal link capacity C. The network is further modeled as a directed graph $G = (V, E)$, where there exists a link $(u, v) \in E,\ u, v \in V$ if the distance between the nodes, $d(u, v)$, is less than the (maximum) transmission range R. Two links a and e interfere with each other if

$$d(t(a), r(e)) \leq R(a) \vee d(r(a), t(e)) \leq R(e), \qquad (1)$$

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. This interference model is referred to as the Boolean interference model which says that a node is only capable of receiving a transmission if it is inside the transmission radius of only one active node.

In a wireless network, not all links can be active simultaneously due to interference, and thus, we have to establish a schedule $\boldsymbol{\alpha}$ which tells us how the links are used. All the links that are active simultaneously have to belong to the same independent set of links to avoid collisions.

The independent sets that are used for transmitting are referred to as transmission modes, and we denote the set of transmission modes with $\mathcal{M} = \{\mathcal{L}_1, \ldots, \mathcal{L}_M\}$. The schedule $\boldsymbol{\alpha} = \{t_1, \ldots, t_M\}$ assigns each transmission mode $\mathcal{L}_i$ with the proportion of time t_i that it is used. Now the effective capacity of link e is

$$c(e) = C \sum_{i=1}^{M} t_i \mathbf{1}_{\{e \in \mathcal{L}_i\}}, \tag{2}$$

that is, the nominal capacity times the time share the link is active.

3.2 Forwarding capacity

The forwarding capacity I^* is defined as the maximum sustainable density of packet flow [1/m/s], i.e., the number of packets crossing a unit length of a line perpendicular to the flow per unit time. Alternatively, I^* can be interpreted to represent the maximum sustainable mean density of progress [22], i.e., the density of packets times their mean velocity in a given direction. The maximum sustainable density of flow (obtained with optimal global coordination of the transmissions) depends on the physical parameters at hand: density of nodes λ [1/m^2], (maximum) transmission range R [m], and nominal capacity of a link C [1/s]. By dimensional analysis [3], I^* can be expressed as any combination of the parameters having the dimension 1/m/s times a function of all the independent dimensionless parameters that can be formed. A combination of parameters of dimension 1/m/s is provided by $C\sqrt{\lambda}$, and there is only one dimensionless parameter, namely the mean degree of a node $\nu = \lambda \pi R^2$ (the constant π is unimportant as it can be absorbed in the definition of u below). Thus,

$$I^*(\nu) = C\sqrt{\lambda}\, u^*(\nu), \tag{3}$$

where u^* is an unknown function referred to as the dimensionless mean progress and is used to describe the forwarding capacity when the physical parameters C and λ are not fixed. Achievable forwarding capacity I, that is attained with a specific scheme for coordinating the transmissions, is a lower bound for I^*.

4. PROBLEM DECOMPOSITION

This section ties the problem of finding the forwarding capacity I^* of an infinite wireless multihop network into a broader problem of maximizing the total data flow in dense multihop networks. A dense network corresponds to a network in a closed domain with the nodes having an infinitesimal transmission range, and the paths being smooth geometric curves allowing a continuous representation of the network. In the limit of a dense network [10, 11], maximizing the capacity of the network separates into two problems, as detailed below.

4.1 Separation of scales

Given a traffic demand density profile, i.e., the traffic demand between every two points of the network, and a set of paths $\mathcal{P}$, we can calculate the traffic load, $\Phi(\boldsymbol{r}, \mathcal{P})$, at position $\boldsymbol{r}$. This can be written as a fraction of the total rate of the packet flow, $\Phi(\boldsymbol{r}, \mathcal{P}) = \Lambda\Psi(\boldsymbol{r}, \mathcal{P})$. Assuming that in a single time slot packets are forwarded to only one direction, the achieved forwarding capacity, I, sets the limit, $\Lambda\Psi(\boldsymbol{r}, \mathcal{P}) \le I$ for all $\boldsymbol{r}$, for the local traffic load. Hence, $\Lambda \le I / \max_{\boldsymbol{r}} \Psi(\boldsymbol{r}, \mathcal{P})$, and in order to maximize the capacity of the network, i.e., solve $\max \Lambda$, we have two separate problems: 1) maximize the achieved forwarding capacity, $I^* = \max I$, and 2) minimize the maximal load.

Minimizing the maximal load (problem 2 above) corresponds to load balancing and is referred to as the macroscopic problem. Determining the forwarding capacity I^* (problem 1 above) represents the microscopic level problem capturing the properties of the underlying wireless network from the point of view of a single node. Note that for a given domain and a given traffic demand profile the maximum local load is a given constant. Thus, the scaling law $O(1/\sqrt{n})$ of the network capacity per node [9] follows trivially from this relation together with (3).[1]

4.2 Microscopic level problem

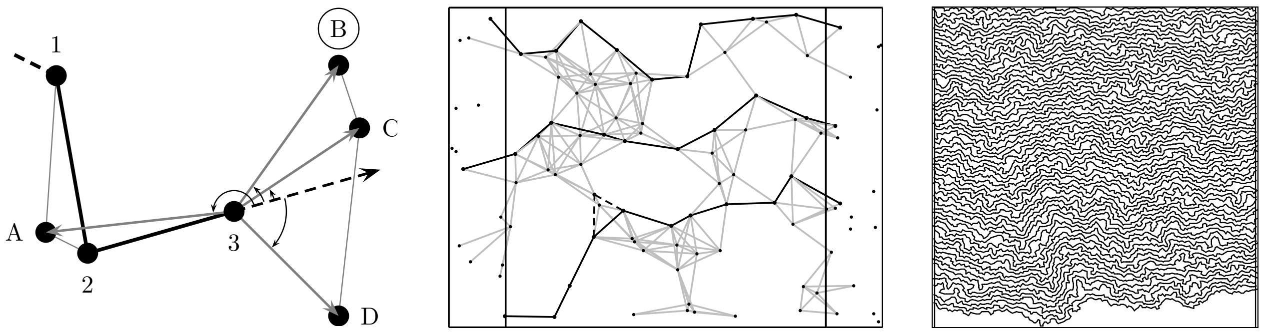
The focus of this paper is on the microscopic level problem, i.e., finding I^*. At the microscopic level, the assumption of a dense network implies that from the local perspective the network appears as an infinite network. Two randomly selected nodes are, on the average, much further apart from each other than two neighboring ones. If the nodes communicating with each other are assumed to be random, a route between a source and a destination typically consists of a large number of hops. Therefore, the amount of relay traffic in a specific area of the network is much higher than the amount of traffic that originates from or terminates to the area.

When considering the relay traffic, we do not require full connectivity. The nodes that do not belong to the transport network can be connected to it via an access network in negligible time, since the amount of originating and terminating traffic in an infinite network is very small compared to the amount of relay traffic, cf. [8].

It is assumed that the forwarding capacity is shared between different directions using appropriate scheduling. The direction distribution in different areas of the network might be different. This mismatch becomes negligible when the size of the network approaches infinity. Thus, the microscopic level problem considers maximizing the flow of traffic in a given direction.

As a result, no traffic matrix or distribution is needed for determining I^*. We simply search for the maximum amount of traffic that can be relayed through the transport network in a single direction and omit the originating and terminating traffic from the model. In simulations, we consider a finite transport network with a set of artificial sources on one side of the network and a set of artificial sinks on another. The sources generate the relay traffic that actually consists of the packets of multiple OD-pairs with the same

[1]The validity of this is based on the physically intuitive notion of separation of scales at high node densities without a rigorous proof.



(a) Next-hop selection (b) Paths in a small network (c) Paths in a large network

Figure 1: Examples of the operation of the path search algorithm.

direction at this point of the network. The results are finally extrapolated as a function of the network size.

5. PATH SCHEDULING

In this section, we construct an algorithm for finding a densely packed set of paths whose links do not interfere with the links of other paths and that connect the sources and the sinks on the opposite sides of the network. By identifying non-interfering paths, scheduling under our assumed Boolean interference model (1) becomes trivial since every third link in a path can be active. We utilize this idea to obtain a lower bound for the forwarding capacity. The used methods bear similarities to those of geographic routing [21].

With respect to the classical max flow problem, this means that we fix the flow to be zero for the links that are not between the nodes of a path. Thus, when the paths are chosen in the way that, in addition to the previous and following link, a transmission only interferes the reception of the link preceeding the previous one, it is possible schedule a flow of $C/3$ for a single path. This can be done with only three transmission modes.

5.1 Path searching algorithm

The transport network with the average of N nodes resides in a unit square that has a strip of sources added to the left side and another strip of sinks added to the right side. The width of the added strips is R since it is not possible to form links with the nodes of the network from farther away, see Figure 1b.

The algorithm starts from the uppermost source (towards right) and then chooses the leftmost neighbor to be the next hop. The leftmost neighbor is the neighbor that forms the largest angle ($[-\pi, \pi]$) with respect to the current direction of the path. Note, that the neighbors on the right have negative angles. The leftmost neighbor is chosen with one exception: the path is not allowed to cross itself, and such neighbors are ignored. The selection of the next hop is illustrated in Figure 1a. In the figure, we have a path traversing through nodes 1 and 2 to node 3 that has four neighbors (A, B, C, and D) in addition to the used node 2. Continuation to node A forms the largest angle with respect to the current direction of the path (marked with an arrow), but would cause the path to cross itself. Instead, node B is chosen because of the second largest angle. Note that the angle created with node D is negative.

The algorithm continues the path until it reaches an exception. The handling of the exceptions is done in the following way:

- If the path comes to a dead end (no available neighbors), the algorithm returns to the previous node.
- If the algorithm returns to the source, no path from this source can be found. The algorithm continues from the next source.
- If the algorithm reaches another source, then no path from this source and the nodes along the way can be found and they can be ignored later. The algorithm continues from the next source.
- If the algorithm reaches a sink, we have found a path and continue the algorithm from the next source. The nodes belonging to the path and their neighbors are ignored later.

The search is continued until all the sources have been gone through.

The paths that the algorithm finds do not interfere with each other, but it is still possible that the links of the path interfere with other links of the same path in a way that it is not possible to schedule a $C/3$ flow to the path. Nevertheless, it is always possible to pick a subset of the nodes from the original path to form the interference-free path. This can be done by starting from the sink and always choosing the neighbor that has the smallest index in the path to be the previous node in the interference-free path. Now, the formed links do not interfere with each other since the supposedly interfering node would transmit directly to the receiver instead of interfering it. The phenomenon can be seen in the lowest path of Figure 1b, which illustrates the set of paths for a small network, where the algorithm advances using the dashed lines, but the packets need to be transmitted along the solid line.

5.2 Network size

It is easier to connect the sources to the sinks through a small network. For example, in a finite network there is always a positive probability for a non-zero flow, while in an infinite network a positive flow becomes possible only above the percolation threshold ($\nu \approx 4.5$). Thus, the simulations should be done in a network with sufficient number of nodes for the results to be meaningful when considering the infinite network. When the simulations are conducted in a finite network, there are two border effects causing an error to the quantity of interest $\bar{n}_{\text{paths}}$, the density of the paths in the vertical direction, and hence to the achievable forwarding capacity $I = C/3 \cdot \bar{n}_{\text{paths}}$.

The horizontal effect depends on the length of the paths. If the distance between the sources and the sinks is small, the existence of the paths is more probable. When the density of the network, N, (recall that we consider a unit square area) is increased, the capacity (from (3)),

$$u = \bar{n}_{\text{paths}}/(3\sqrt{N}), \tag{4}$$

goes down. The vertical effect is caused by the fact that the top and the bottom of the network confine the paths to a limited area. Thus, the behavior of the paths is different near the borders, and the interesting quantity, the number of paths per height unit, differs from that in the middle of the network. This can be seen in Figure 1c. When the size of the network is increased, the relative area that is wasted due to the artificial limitations decreases, and the capacity goes up.

As can be seen from Figure 2 (solid lines), horizontal effect, dominating with small networks, dies out faster than the vertical effect. Thus, the capacity first goes down rapidly and then slowly starts to increase. The combined effect of the two is that u approaches the asymptotic value from below. It is also possible to calculate the number of paths originating from the middle part of the network, e.g., by adding some extra space on top of and below the network residing in the unit square. In this case, u approaches the asymptotic value from above as can be seen from Figure 2 (dashed lines). The network size used later in the simulations is 10^5.

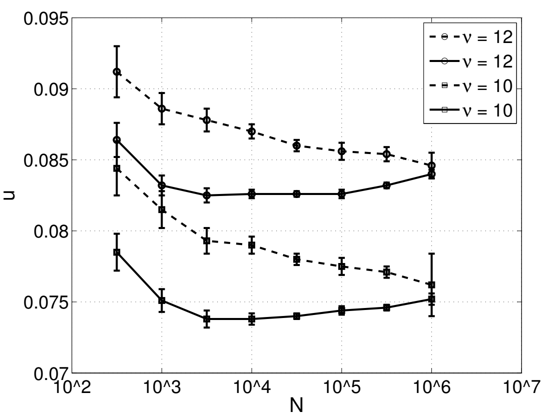


Figure 2: The effect of network size on u using the path search algorithm.

6. GREEDY MAXIMUM WEIGHT SCHEDULING

In this section, we describe another method for resolving the problems related to finding a feasible way to schedule the transmissions with a reasonable outcome. In their paper [24], the authors present a maximum throughput policy, that stabilizes the network for all arrival rates for which it is stabilizable. The original algorithm, often referred to as the backpressure algorithm, allows multiple customer classes and link capacities, but to study the forwarding capacity only one customer class is needed for representing the flow of traffic in one direction.

The algorithm has three stages. In time slot t, the first stage is to calculate a weight $w^t(e)$ for each link $e \in E$ as follows,

$$w^t(e) = q^{t-1}(t(e)) - q^{t-1}(r(e)), \tag{5}$$

where $q(v)$ is the queue length at node $v \in V$. In the second stage, a maximum weight transmission mode is selected

$$\mathcal{L}^*(t) = \underset{\mathcal{L} \in \mathcal{M}}{\arg\max} \sum_{e \in \mathcal{L}} w^t(e). \tag{6}$$

Finally in the third and last stage, if we index the links with $i = 1, \dots, |E|$, we get the activation vector $L(t)$ at time slot t as follows,

$$L_i(t) = \begin{cases} 1, & \text{if } e_i \in \mathcal{L}^*(t), \text{ and } q^{t-1}(t(e_i)) > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

In each timeslot the policy finds the transmission mode that is of the maximum weight. The weight of each link is simply the difference in the queue lengths between the transmitting and receiving end of the link since all the links have the same nominal capacity C. Because every node is only able to participate in one transmission at most, the links in the maximum weight transmission mode are always activated if the transmitter has a packet to send.

If it were possible to run this backpressure algorithm properly, we could find out the true forwarding capacity. Unfortunately, due to limitations in the computing capacity, it is necessary to simplify the problem. We apply the algorithm in a similar setting as we did with the path searching algorithm.

6.1 Application of the method

Again, the transport network of N nodes (on average) resides in a unit square that has a strip of sources added to left side with a fixed queue length of m and a strip of sinks added to the right side. The queue length at sinks is considered always to be zero. The width of the strips is R. The top and the bottom of the network are connected together to form a cylinder and to reduce the border effect in the vertical direction. As a result of a simulation, we get the average number of packets arriving to the sinks in a time slot per unit height, $\bar{n}$, when the network has been stabilized. Since we assume that the packet transmission time matches the time slot duration, it follows that $I = C\bar{n}$, and using (3) the dimensionless mean progress is given by

$$u = \bar{n}/\sqrt{N}. \tag{8}$$

Now, there are three things affecting the result. Firstly, the maximum weight transmission mode is approximated with a maximal transmission mode containing a greedily chosen combination of links with the largest weights. Secondly, the fixed queue length of the sources, m, affects the set of possible link weights and thus the accuracy of the scheduling. The queue length initialization of the other nodes is also important because of the length of the initial transient. Thirdly, the size of the network used in the simulations should be as large as possible to properly represent the infinite network. The effects of these approximations are discussed in the following.

6.2 Greedy scheduling

The main factor limiting the applicability of the original method is the necessity to solve the maximum weight independent set problem since this is typically a complex problem that is only solvable in some special applications. We choose links to the transmission mode in the order of their weight until the independent set of links is maximal. Because of the greedy scheduling, we get a capacity lower than the true capacity.

The performance of the greedy method compared to the true maximum weight independent set in a single-slot case is represented in Figure 3. The figure shows the difference between the greedy instantaneous forwarding capacity and the optimum of [20]. The error made in the case of a continuous flow is unknown.

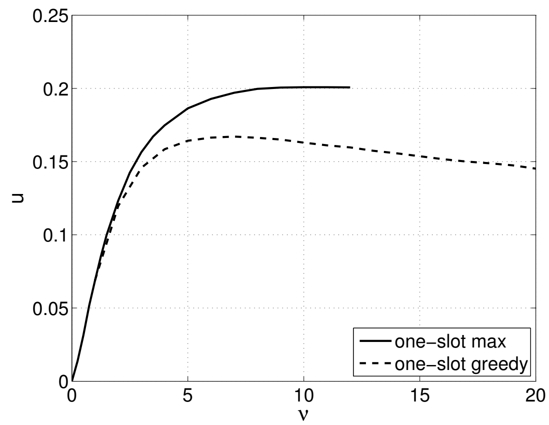


Figure 3: Greedy scheduling compared to the optimum in a single slot situation with fixed transmission radius.

6.3 Queue length initialization

When the number of packets in the network is small, the link weights are also small, and the probability that two links have equal weights is notable. Since the choice between two transmission modes that are of the maximum weight is arbitrary, the operation of the scheduler is, at this point, more random. This leads to suboptimal performance that results in increasing queue lengths. When the queues grow, some of the ties are resolved, and eventually the policy stabilizes the queues for all arrival rates for which it is possible. This might however require very long queues. When the queue length at sources is fixed, the number of possible link weights is also fixed. This means that to assure the best possible performance, the queue length at sources, m, should be as large as possible.

In practice, the choice of the queue length at sources is a compromise since a large m means a long initial transient. Because of the finite queue length at the sources, we get a capacity lower than the true capacity. Figure 4 shows the effect in a thousand node network when the queue length at sources varies from 10 to 1000 packets with $m = 100$ being the value used later on.

6.4 Network size

The size of the network needs to be large enough for the results to be meaningful when considering a network with

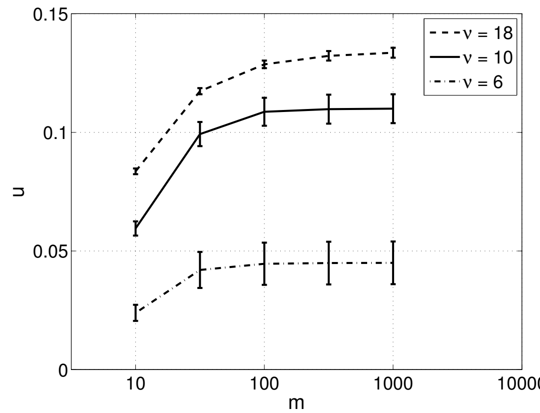


Figure 4: The effect of the queue length of the sources with fixed transmission radius.

infinite size. A small network gives too high values for the achievable forwarding capacity since it is easier to establish a flow through shorter paths.

Because of the finite network size, we get a capacity higher than the true capacity. By eliminating the effect of the limited network size, it would be possible to study capacities that are actually achievable with greedy maximum weight scheduling. Figure 5 represents u as a function of the network size N with $m = 100$. As can be seen, the network size $N = 1000$ used in the final simulations is not quite enough when ν is small. When the network is more homogeneous from the scheduling point of view ($\nu = 18$), the reduction in u is mainly due to the fact that $m = 100$ is not enough for a large network (cf. Figure 4).

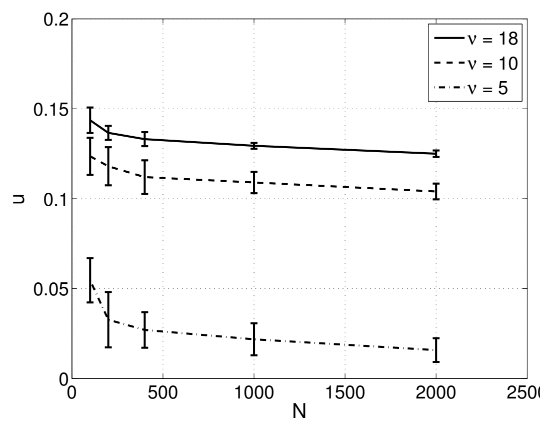


Figure 5: The effect of the network size in greedy maximum weigth scheduling with fixed transmission radius.

7. RESULTS

The simulations were conducted with the following kind of setup. The nodes were distributed to a rectangular network area with width $1 + 2R$ and a varying height according to a spatial Poisson point process with density N. The nodes within one transmission radius from the left side of the network were assigned the role of a source and the ones within R from the right side of the network were sinks. The relay

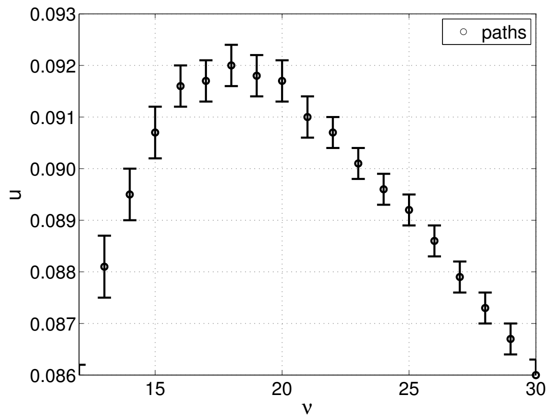


Figure 6: The dimensionless mean progress as a function of the mean neighborhood size using path search algorithm.

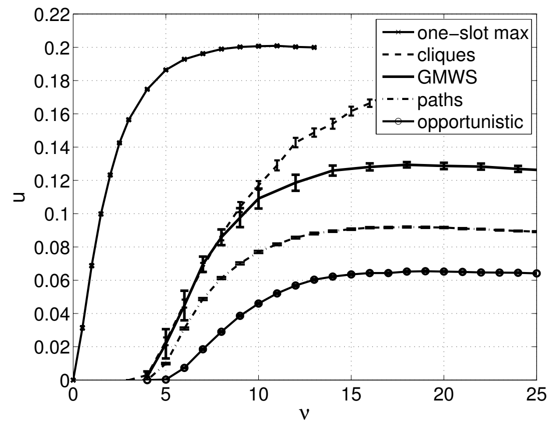


Figure 7: Comparison of the results. The curves of the figure are in the same order as in the legend.

network was a unit square with the average of N nodes in the middle of the network. When N is large enough the results meaningfully represent the infinite network that appears when a massively dense network is viewed from the perspective of a single node.

7.1 Path scheduling

The path search algorithm gives as a result the number of independent paths connecting the sources to the sinks. With the used network height of 3, the paths beginning from a source with the y-coordinate between 1 and 2 were used as an estimate for the vertical path density $\bar{n}_{\text{paths}}$. From this, we can calculate the dimensionless mean progress with (4).

Figure 6 presents u as a function of ν near the optimum size of the neighborhood (see also Fig. 7 for wider range) with fixed transmission radius. The maximum of $u = 0.09$ and occurs at $\nu = 18$. The results have been obtained with $N = 10^5$, and are averages over 100 network realizations. The errorbars show the 95 % confidence intervals.

The advantage of the algorithm is that it is computationally efficient, and the simulations can be done with a very large network. This is important when the network appears heterogeneous from the point of view of a single node, i.e., ν is small. Thus, the results of path scheduling for the smallest values of ν are very accurate. The paths connecting the sources to the sinks occur so rarely that routing is only possible along these paths, and different paths do not interfere with each other.

7.2 Greedy maximum weight scheduling

The simulation setup with the greedy maximum weight scheduling differs from the one of the path scheduling in the way that the top and the bottom of the network of unit height are connected together to form a cylinder. This is done to reduce the border effects in the vertical direction. The greedy maximum weight scheduling algorithm gives as a result the number of packets per slot arriving to the sinks, and the dimensionless mean progress can be obtained with (8). Alternatively, it is possible to monitor the number of packets leaving the sources or the progress of the packets in the horizontal direction. When these three quantities are approximately the same, the simulation can be considered to have reached steady state.

Figure 7 presents the results obtained from the greedy maximum weight scheduling compared to other bounds for the forwarding capacity with fixed transmission radius. The results have been obtained with $m = 100$ and $N = 1000$. They are averages over 10 network realizations and the errorbars show the 95 % confidence intervals. The values of m and N are a practical compromise between the accuracy and the necessity to keep the simulation times reasonable. The maximum, $u = 0.13$, occurs at $\nu = 18$. Since neither of the studied methods is particularly good at coordinating the transmission when the mean degree of a node is large, the true optimum size for ν is likely to be higher.

When ν is very small, the difference between the results of greedy maximum weight scheduling and path scheduling is due to the network size. The one third flow that the scheduling is able to achieve along a single path is maximal so the results do not suffer from the greedy heuristic with the smallest neighborhoods. The smaller network used with greedy maximum weight scheduling leads to too high capacity as there is more likely to be better connectivity through a small network.

When ν is larger, the effect of the network size is less relevant since the network is more homogeneous from the scheduling point of view. That is to say, there are always multiple possible links to choose from, and no clear bottlenecks appear as the network is made larger. The capacities are achievable since the meaningful effects, the greedy scheduling and the queue length at sources, both move the result downwards. Greedy scheduling is not able to coordinate the transmissions efficiently enough when the number of interfering links is large, and, in addition to the weight, one should also consider how the links interfere with each other. Also, the queue length at sources should be made larger when the number of links grows to allow the weights of the links to be separable.

7.3 Comparison with other methods

Next, we present some previous results that are also illustrated in Figure 7 for comparison. The one-slot maximum [20] represents the maximum forwarding capacity that can be achieved in a single time slot. It is an upper bound and cannot be reached with continuous flow since it is not possible to use the corresponding transmission mode constantly.

The asymptotic analysis made in [20] for the instantaneous capacity shows also that there is a neighborhood size that is optimal w.r.t. u as the capacity goes down when ν tends to infinity.

Of the results for continuous flows, the clique curve [18] is obtained by solving a classical max flow problem for a reduced set of necessary clique constraints in random network realizations. It is also an upper bound. The lowest curve, on the other hand, corresponds to a random access forwarding method called opportunistic forwarding that is similar to ExOR [2]. In opportunistic forwarding, all nodes with packets transmit with a fixed probability, and from the neighbors able to receive the transmission, the one with the greatest progress is chosen to be the next hop. As it is an actual forwarding method, opportunistic forwarding gives a lower bound for the forwarding capacity.

Compared with these, the new results give a first reasonable estimate for the forwarding capacity. For example at $\nu = 10$ where GMWS already gives a lower bound,[2] the gap between the upper bound (clique curve) and the lower bound is small. Considering the larger values of ν, the gap is reduced at least by half.

7.4 Power control

In addition to a fixed transmission radius, we consider greedy maximum weight scheduling in a case where the transmission radius $R(e)$ can be reduced from the maximum value R to the length of link e. The idea is that the transmitting nodes use a radius just large enough to reach the receiving node and thus minimize the interference. In this case, the parameter ν corresponds to the mean number of neighbors with the maximum radius.

Figure 8 shows $u(\nu)$ for greedy maximum weight scheduling with both fixed and adjustable transmission radius. With power control, it is possible to activate more links simultaneously, and the maximum gain from a freely adjustable transmission radius is a little over 50 %.

8. CONCLUSIONS

In this paper, we studied the forwarding capacity that is achievable in a random geometric network under the Boolean interference model. By using two constructive approaches (the path scheduling and the greedy maximum weight scheduling), we were able to obtain new numerical results on the forwarding capacity as a function of the mean node degree. The results were also studied as a function of the network size to allow us to extrapolate the results to an infinite plane.

In the first approach, we obtained a lower bound for the forwarding capacity using an algorithm that searches for independent paths through the network. Using these paths, the usually challenging scheduling problem becomes easy allowing us to study the capacity with very large networks. In the second approach, we applied maximum weight scheduling where the scheduling in a time slot was solved using a greedy heuristic. This provided a practical approach for our simulations allowing us to consider reasonably large networks and to produce what we believe are the most accurate numerical results that are presently available on the forwarding capacity. Together with the existing results to the load balancing problem [11], the estimate for the forwarding capacity can be used to calculate the overall capacity of a dense wireless multihop network under the used assumptions.

[2] GMWS is not a lower bound with the smallest values of ν as the effect of the network size has not been fully eliminated.

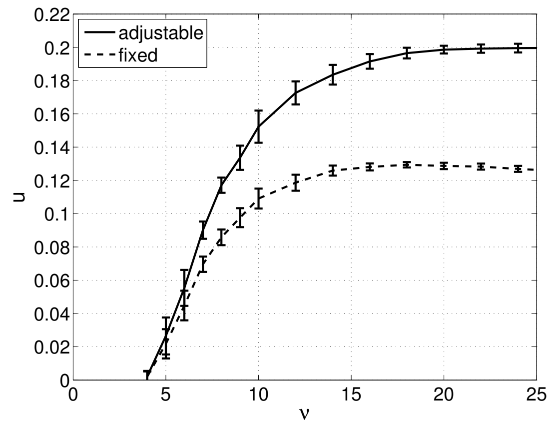


Figure 8: Function $u(\nu)$ for greedy maximum weight scheduling with and without power control.

In addition to an even more accurate value for the forwarding capacity, the exact size of the optimal neighborhood remains as an open question for future research. Even so, the growth in the capacity is very modest after a certain point, and a more practical line of future research would be to study the distributed implementation of greedy maximum weight scheduling or the impact of other, more realistic, types of interference models.

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.

9. 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] S. Biswas and R. Morris. ExOR: opportunistic multi-hop routing for wireless networks. *SIGCOMM Computer Communications Review*, 35(4):133–144, 2005.

[3] E. Buckingham. On physically similar systems; illustrations of the use of dimensional equations. *Phys. Rev.*, 4(4):345–376, 1914.

[4] R. Catanuto, S. Toumpis, and G. Morabito. Opti{c,m}al: Optical/optimal routing in massively dense wireless networks. In *Proceedings of IEEE INFOCOM*, pages 1010–1018, 2007.

[5] P. Chaporkar, K. Kar, and S. Sarkar. Throughput guarantees through maximal scheduling in wireless networks. In *Proc. of 43rd Annual Allerton Conference on Communication, Control, and Computing*, 2005.

[6] O. Dousse, F. Baccelli, and P. Thiran. Impact of interferences on connectivity in ad hoc networks. *IEEE/ACM Transactions on Networking*, 13(2):425–436, 2005.

[7] A. Eryilmaz, O. Asuman, and E. Modiano. Polynomial complexity algorithms for full utilization of multi-hop wireless networks. In *Proceedings of IEEE INFOCOM*, pages 499–507, 2007.

[8] 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.

[9] P. Gupta and P.R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388–404, 2000.

[10] E. Hyytiä and J. Virtamo. On load balancing in a dense wireless multihop network. In *NGI 2006, 2nd Conference on Next Generation Internet Design and Engineering*, pages 72–79, 2006.

[11] 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.

[12] P. Jacquet. Geometry of information propagation in massively dense ad hoc networks. In *MobiHoc '04: Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*, pages 157–162, 2004.

[13] K. Jain, J. Padhye, V. Padmanabhan, and L. Qiu. Impact of interference on multi-hop wireless network performance. In *Proceedings of ACM MobiCom*, pages 66–80, 2003.

[14] M. Kalantari and M. Shayman. Energy efficient routing in wireless sensor networks. In *Conference on Information Sciences and Systems*, 2004.

[15] M. Kodialam and T. Nandagopal. Characterizing achievable rates in multihop wireless mesh networks with orthogonal channels. *IEEE Transactions on Networking*, 13(4):868–880, 2005.

[16] X. Lin and N. B. Shroff. The impact of imperfect scheduling on cross-layer rate control in wireless networks. In *Proceedings of IEEE INFOCOM*, 2005.

[17] M.J. Neely. Dynamic power allocation and routing for time varying wireless networks. *IEEE Journal on Selected Areas in Communications*, 23:89–103, 2005.

[18] J. Nousiainen and P. Lassila. Approximating maximum directed flow in a large wireless network. In *IEEE ICC*, Dresden, Germany, 2009.

[19] J. Nousiainen, J. Virtamo, and P. Lassila. Forwarding capacity of an infinite wireless network. In *Proceedings of ACM MSWiM*, pages 177–184, Vancouver, CA, 2008.

[20] J. Nousiainen, J. Virtamo, and P. Lassila. Maximum weight independent sets in an infinite plane. In *ITC21*, Paris, France, 2009.

[21] I. Stojmenovic. Position based routing in ad hoc networks. *IEEE Communications Magazine*, 40(7):128–134, 2002.

[22] H. Takagi and L. Kleinrock. Optimal transmission ranges for randomly distributed packet radio terminals. *IEEE Transactions on Communications*, 32(3):246–257, 1984.

[23] L. Tassiulas. Linear complexity algorithms for maximum throughput in radio networks and input queued switches. In *Proceedings of IEEE INFOCOM*, pages 533–539, 1998.

[24] 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.

[25] X. Wu, R. Srikant, and J. R. Perkins. Scheduling efficiency of distributed greedy scheduling algorithms in wireless networks. *IEEE Transactions on Mobile Computing*, 6(6):595–605, 2007.