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

Department of Communications and Networking
Helsinki University of Technology
P.O.Box 3000, FIN-02015 TKK, Finland

ABSTRACT

We study the maximal forwarding capacity of a massively dense wireless multi-hop network where a typical path consists of a vast number of hops. In such a network, the macroscopic level, corresponding to the scale of an end-to-end path, and the microscopic level, corresponding to the scale of a single hop, can be separated. At the macroscopic level the task is that of routing, while at the microscopic level the packets are forwarded based on the information received from the macroscopic level. We give a formulation for the forwarding problem and devise simulation algorithms based on an augmentation of the max-flow min-cut theorem for obtaining upper bounds for the maximal forwarding capacity. We compare the upper bounds with feasible forwarding methods and find out that the tightest bound is about three times the highest achieved performance.

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

Wireless multi-hop networks, where nodes communicate with each other over several intermediate hops, have received a lot of attention in the recent years in the context of ad hoc, mesh, and sensor networks. In principle, using multihop communications enables, e.g., enlarging the coverage area. However, to be able to design efficient mechanisms for exploiting these capabilities requires a careful analysis of the impact of the wireless interference phenomena on the capacity of wireless multihop networks.

We consider dense (or large) wireless multihop networks, similarly as in [4, 8, 10]. So-called massively dense networks arise as part of, e.g., large-scale sensor networks where thousands of sensor nodes may communicate over short-range radios to form a single connected network operating as a part of future ubiquitous networks.

When the network is very large, the macroscopic level, corresponding to the scale of an end-to-end path, and the microscopic level, corresponding to the scale of a single hop, can be separated. The macroscopic level routing protocol, treating the network as a continuous medium, provides the direction of packet flow to the microscopic level where the packets are forwarded based on this information according to the rules of the microscopic level forwarding method. Considering one direction at a time, there exists a certain maximum flow of packets that can be supported. The maximum flow can be divided between different directions, e.g., via time sharing.

In this paper, we obtain an upper bound for the microscopic level forwarding capacity. As the obtained results are upper bounds, we do not consider specific strategies to achieve certain capacities. We model the network as a random geometric graph (with infinite size) where each node has a single radio transmitter and interference is modeled by the Boolean interference model. By assuming a slotted time system, the capacity of the network can be characterized by an augmented max-flow min-cut theorem where the transmission radius of a node and the transmission schedule are additional parameters that need to be optimized.

Direct application of the max-flow min-cut theorem is not feasible due to the prohibitive size of the network. However, upper bounds may be obtained by limiting the size of the set of cuts. In particular, we restrict the set of cuts to consist of only straight lines and consider three cases providing increasingly tighter upper bounds, namely 1, 2, and infinite number of cuts. We show that each of these formulations can be translated into a corresponding optimization problem related to determining the size (or weight) of the maximal independent set of links. Notably, we develop a novel simulation based algorithm, the Moving Window Algorithm (MWA), for iteratively constructing the weight per unit length (or unit area) of the independent set of links crossing the given cut sets (consisting of infinitely long cuts).

Additionally, we have experimented with an existing forwarding method, namely the ExOR protocol [3], with some

MSWiM'08, October 27–31, 2008, Vancouver, BC, Canada.

local coordination between the nodes. By simulating its performance under heavy traffic, we can obtain feasible lower bounds for the forwarding capacity.

Our results demonstrate that there is still a difference by a factor of 3 between the tightest (infeasible) upper bound and the highest achievable lower bound. This comparison gives us insight into the possible benefits of more global coordination of the network.

In the considered setting, the exact value of the maximal flow of packets at the microscopic level is not known. A seminal result in this field is given in [7] where the asymptotic result is established that in networks with arbitrarily located nodes the capacity per node scales as $O(1/\sqrt{n})$, where n is the number of nodes. For randomly located networks it has been shown recently in [2] and [6] that the same scaling behavior can be achieved with a carefully designed transmission scheme. As pointed out in [6], the exact capacity (the constant in front of the scaling law) is still not known, and in this paper we provide some partial results for this in a specific scenario. The max-flow min-cut theorem has been applied in wireless networks, e.g., in [5] where a feasibility condition yielding an upper bound for the solution is established.

The notion of mean progress to capture the microscopic level performance was defined in [11]. However, the analysis only considered the local performance of the network (i.e., from the point of view of a single node) under a simple forwarding strategy for different wireless access methods. Simulation results on the density of progress for various forwarding schemes in the network setting have been given in [1].

The occurrence of the separation of spatial scales between the microscopic and macroscopic levels has been pointed out in, e.g., [8]. Specifically, [8] analyzes the macroscopic level problem where the task is to determine the continuous routing paths so that the maximum local load in the network is minimized. Related results to this problem can also be found in [4, 10].

The rest of the paper is organized as follows. In Section 2 we present the problem decomposition into two subproblems, one of which is considered in this paper. Section 3 defines the general framework that includes the assumptions and the used network model. In Section 4 a formulation is given for the problem, and three cases are taken under consideration. The simulation algorithms for obtaining an upper bound for the forwarding capacity in these cases are presented in Section 5. The results from the simulations are given in Section 6, while Section 7 concludes the paper.

2. PRELIMINARIES

2.1 Problem decomposition

In a large wireless multihop network, a path between a sender and a receiver consists of a large number of hops. In the limit of a dense network, see, e.g., [9], the paths are smooth geometric curves allowing a continuous representation of the network, and maximizing the capacity of the network separates into two problems, as detailed below.

Adopting the model from [9], a dense network corresponds to a network in a closed domain with nodes having an infinitely short transmission range. The traffic demand density profile $t(\boldsymbol{r}_1, \boldsymbol{r}_2)$ [1/m^4] is the fraction of the total rate of packet flow Λ that originates from a differential area element in $\boldsymbol{r}_1$ and is destined to a differential area element in $\boldsymbol{r}_2$. It holds that

$$\iint t(\boldsymbol{r}_1, \boldsymbol{r}_2) d^2\boldsymbol{r}_1 d^2\boldsymbol{r}_2 = 1. \quad (1)$$

With a given traffic profile and a set of paths $\mathcal{P}$, the traffic load at $\boldsymbol{r}$ equals[1]

$$\Psi(\boldsymbol{r}, \mathcal{P}) = \lim_{d \to 0} \frac{t_{\mathcal{D}}}{d}, \quad (2)$$

where $t_{\mathcal{D}}$ is the fraction of packets that traverses through a disk $\mathcal{D}(\boldsymbol{r}, d/2)$ of diameter d at point $\boldsymbol{r}$. Forwarding capacity I sets the limit

$$\Lambda\Psi(\boldsymbol{r}, \mathcal{P}) \leq I \; \forall \, \boldsymbol{r} \Rightarrow \Lambda \leq \frac{I}{\max_{\boldsymbol{r}} \Psi(\boldsymbol{r}, \mathcal{P})} \quad (3)$$

for the local traffic load, and in order to maximize the capacity of the network, i.e., solve $\max \Lambda$, we have two separate problems: 1) maximize the 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 maximal 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.

2.2 Microscopic level problem

In this paper we focus on the problem of determining 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. This allows us to concentrate purely on the relay traffic and omit the originating and terminating traffic from the model. No traffic matrix or distribution is needed for determining I^*, but we simply maximize the amount of traffic that can be relayed through the network. In a network there is traffic flowing in different directions. However, assuming that time is slotted the different directions can be treated independently so that the maximal forwarding capacity I^* can be shared between the different directions using appropriate scheduling. Thus, the microscopic level problem considers maximizing the flow of traffic in a given direction.

3. NETWORK MODEL

3.1 Assumptions

The system is assumed to be synchronous, and time is assumed to be divided into slots. The nodes are assumed to be static, reliable, and located according to a spatial Poisson point process in two dimensions. The intensity of the process, referred to as the node density, is denoted by λ [1/m^2].

[1]The corresponding notations in [9] are $\lambda(\boldsymbol{r}_1, \boldsymbol{r}_2) = \Lambda t(\boldsymbol{r}_1, \boldsymbol{r}_2)$ and $\Phi(\boldsymbol{r}, \mathcal{P}) = \Lambda\Psi(\boldsymbol{r}, \mathcal{P})$.

The nodes communicate with each other over the same frequency of a wireless medium using an omnidirectional antenna and a common fixed transmission power, resulting in a transmission range R [m], and a constant nominal link capacity[2] C [1/s] independent of R. This means that there is a link between two nodes if they are within the distance R from each other, and if the nodes correspond to the vertices, V, of a directed graph $G = (V, E)$, there exists an edge $(u, v) \in E,\ u, v \in V$ if $d(u, v) \leq R$, where $d(u, v)$ is the Euclidean distance between the nodes. The density of the network can be described with the mean degree (mean number of neighbors) of a node $N_R = \lambda\pi R^2$ which is dimensionless.

Simultaneous transmissions interfere with each other, and the effect of interference is described with the interference area of a link. The interference area $\mathcal{I}(e)$ is the set of links that affect or are affected by the use of link $e \in E$. If link e is currently active, the attempt to activate any other link in $\mathcal{I}(e)$ will result in a collision. The interference model we use to model collisions is the Boolean interference model, according to which a node is only able to receive a packet if it hears exactly one transmission inside its transmission radius including its own, and thus

$$\mathcal{I}_B(e) = \{a \in E | d(t(a), r(e)) \leq R \vee d(r(a), t(e)) \leq R\}, \quad (4)$$

where $t(e)$ is the transmitting node of link $e \in E$ and $r(e)$ the receiving node.

With the term network we refer to a pair (G, c) of a graph and a mapping $c : E \to \mathbb{R}^+$. In general, $c(e)$ is called the capacity of the edge e. In a wireless network, not all links can be active simultaneously due to interference, and thus, in our model for a given link e, the effective link capacity $c(e)$ is less than the nominal capacity C. To define $c(e)$, 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. A set of links $\mathcal{L}$ is said to be independent if

$$\forall a \neq e\ :\ a \notin \mathcal{I}_B(e),\ a, e \in \mathcal{L}. \quad (5)$$

We call the independent sets[3] that are used for transmitting transmission modes and denote the set of transmission modes with $\mathcal{M} = \{\mathcal{L}_1, \ldots, \mathcal{L}_n\}$. The schedule $\boldsymbol{\alpha} = \{t_1, \ldots, t_n\}$ 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

$$c(e) = C \sum_{i=1}^{n} t_i \mathbf{1}_{\{e \in \mathcal{L}_i\}}, \quad (6)$$

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 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 [11], 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], transmission range R [m], and nominal capacity of a link C [1/s]. By dimensional analysis, 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. There is a combination of parameters of dimension 1/m/s, namely $C\sqrt{\lambda}$, and only one dimensionless parameter, namely $N_R = \lambda\pi R^2$ (the constant π is unimportant as it can be absorbed in the definition of u). Thus,

$$I^* = C\sqrt{\lambda}\, u(N_R), \quad (7)$$

where u is an unknown function. There is a given value N_R^* that maximizes the function u. Our task is to find this N_R^* (or, equivalently, for a given λ the optimal transmission range R^*) as well as the maximum u^*. The maximal forwarding capacity is then finally $I^* = C\sqrt{\lambda}\, u^*$.

Note that the famous scaling law $O(1/\sqrt{n})$ of the network capacity per node mentioned in Section 1 follows trivially from this relation together with (3), noting that for a given domain and a given traffic demand profile $t(\boldsymbol{r}_1, \boldsymbol{r}_2)$ the denominator of (3) is a given constant. However, the validity of (3) is based on the physically intuitive notion of separation of scales at high node densities, for which we have not given a rigorous proof.

3.3 Flow network definitions

In this paper results from flow networks will be applied to obtain bounds for I^*, and this requires some additional notation to be introduced.

Let us now consider a network (G, c) where we distinguish two special vertices: the start node s and the terminal node t, or the source and the sink, such that t is accessible from s. Accessibility means that there exists a sequence of vertices $(v_0, \ldots, v_n)$ (a walk) such that $(v_{i-1}, v_i) \in E$ for $i = 1, \ldots, n$, and $v_0 = s$ and $v_n = t$. Now, we have a structure $\mathcal{N} = (G, c, s, t)$ that we call a flow network, and we can define a flow in the network. A mapping $f : E \to \mathbb{R}^+$ is a flow if it satisfies the following conditions:

1. $f(e) \in [0, c(e)] \quad \forall\, e \in E$
2. $\sum_{r(e)=v} f(e) = \sum_{t(e)=v} f(e) \quad \forall\, v \in V \setminus \{s, t\}$

The first, feasibility condition guarantees that there is a positive (≥ 0) bounded flow through every arc, and the second, flow conservation conditions means that flows are preserved (except at the source and the sink). The value of flow f is

$$w(f) = \sum_{t(e)=s} f(e) - \sum_{r(e)=s} f(e) = \sum_{r(e)=t} f(e) - \sum_{t(e)=t} f(e). \quad (8)$$

Before representing one of the fundamental results in flow theory, one definition is still needed. A cut of $\mathcal{N}$ is a partition $V = S \cup T,\ S \cap T = \emptyset$ such that $s \in S$ and $t \in T$. The capacity of the cut (S, T) is

$$c(S, T) = \sum_{t(e) \in S,\ r(e) \in T} c(e). \quad (9)$$

The cut $Q = (S, T)$ is called minimal if $c(S, T) \leq c(S', T')$ for all cuts of the network. The capacity of the minimum cut has a significant effect on the capacity of the network.

[2] We have chosen to measure the capacity of a link in terms of packet flow per unit time (instead of bit/s) in order to conform with how traffic load (2) has been defined in [8, 9]. This choice is inconsequential for the results of the paper.

[3] It is enough to consider maximal ones.

4. MAXIMUM FLOW PROBLEM IN WIRELESS NETWORKS

The maximum flow problem is a classic problem in graph theory and combinatorial optimization with a variety of applications. It considers finding a feasible flow through a flow network that is maximal. A flow f is maximal if $w(f) \geq w(f')$ for all flows f' on $\mathcal{N}$. The following max-flow min-cut theorem considers the duality of this problem:

(Ford and Fulkerson 1956) *The maximal value of a flow on a flow network $\mathcal{N}$ equals the minimal capacity of a cut in $\mathcal{N}$.*

This basically means that the bottlenecks of the network dictate the amount of traffic the network can carry.

With a fixed schedule, the maximal flow in the wireless flow network equals the capacity of the minimal cut, but to find the overall maximum value for the flow, we also have to optimize the schedule. Additionally, to find the optimal N_R, maximization with respect to R is also required with a fixed λ. The value of the optimal flow thus ensues from the problem

$$\max_R \max_{\boldsymbol{\alpha}} \min_{Q \in \mathcal{Q}} c(Q, \boldsymbol{\alpha}), \tag{10}$$

where $c(Q, \boldsymbol{\alpha})$ is the capacity of cut Q with schedule $\boldsymbol{\alpha}$ that depends on R through the transmission modes $\mathcal{M}$, and $\mathcal{Q}$ is the set of all cuts. The task quickly becomes infeasible, as the size of the network grows.

According to (10), the maximum with fixed R can be obtained by maximizing the minimum capacity of a cut with respect to the schedule. We can get an upper bound for the performance by limiting our examinations to a smaller set of cuts $\mathcal{Q}' \subset \mathcal{Q}$, because the minimum of a subset is always greater than or equal to the original minimum. This gives us constraint

$$\begin{aligned} w(f_R^*) &= \max_{\boldsymbol{\alpha}} \min_{Q \in \mathcal{Q}} c(Q, \boldsymbol{\alpha}) \\ &\leq \max_{\boldsymbol{\alpha}} \min_{Q \in \mathcal{Q}'} c(Q, \boldsymbol{\alpha}) \quad \forall \mathcal{Q}' \subset \mathcal{Q} \end{aligned} \tag{11}$$

for the maximum value of the flow.

In this paper we study three special cases for the selection of $\mathcal{Q}'$, namely 1, 2, and an infinite number of straight cuts. This yields increasingly tighter upper bounds for the maximal flow.

Since we only examine relay traffic, for instance from left to right, the source(s) and sink(s) are not explicitly visible, and the performance is measured in terms of I. With straight cuts this does not cause any complications, and, naturally, the maximum flow also maximizes the progress per area and time.

4.1 One cut

Because the forwarding capacity, I^* as given by (7), can be interpreted as the number of packets crossing a unit length of line perpendicular to the direction of the flow in unit time (capacity per length), it is reasonable to consider cuts that correspond to a straight line in the vertical direction as the limited set of cuts $\mathcal{Q}'$ in (11) when the traffic is flowing in the horizontal direction. If the limited set of cuts consists solely of a single cut, the task equals finding the size of the maximum independent set of links crossing the given line. This follows from the fact that each independent set of links has a certain number of links crossing the cut, and maximizing with respect to $\boldsymbol{\alpha}$ assigns all the time for the independent set(s) with the greatest number.

The idea for approaching the problem is to use simulations. By obtaining a value for the size of the maximum independent set per unit length, we can estimate the capacity of the network. Since $I^* = Cn/L$, where n is the number of links crossing the length L of line, the dimensionless mean progress, $u = u(N_R)$, can be approximated with

$$u \approx \frac{1}{C\sqrt{\lambda}} \cdot \frac{Cn}{L} = \frac{n\sqrt{\pi}}{\sqrt{N_R}\, L/R}. \tag{12}$$

The simulation algorithm will be introduced in Section 5.

4.2 Two cuts

Considering just one cut gives a relatively loose upper bound for the forwarding capacity since such a performance can only be achieved very locally. After the maximization with respect to $\boldsymbol{\alpha}$ is done, the cut does not represent an average cut (and much less a minimum cut), since the same kind of performance cannot be achieved with a cut that suffers from the interference caused by the links of this cut. By considering two vertical cuts simultaneously, it is possible to obtain a tighter upper bound for $u(N_R)$.

When the number of cuts is two, the task is to maximize the smaller of the capacities of the two cuts, denoted by c_1 and c_2. It is obvious that $\min\{c_1, c_2\} \leq (c_1 + c_2)/2$. Let us now consider the independent set of links that maximizes the number of times that the links cross the two lines. We assert that a schedule using only this independent set maximizes the sum in the above inequality. Because the situation is symmetric, and the lines are infinitely long, both lines are crossed equally many times, i.e., $c_1 = c_2$. As a result, the left hand side and the right hand side are equal, and as the right hand side was maximized, we have certainly obtained the maximum for the left hand side. The schedule is thus optimal.

Because we want the effect of horizontal interference to be included, the choice of the distance between the two cuts needs to be done carefully. If the cuts are far from each other, they are independent, and the result is the same as with one cut. On the other hand, if the cuts are very close to each other, almost all the links cross both of the lines, and because the links crossing both the lines are counted twice, the result is nearly the same again. Thus, in order to maximize the effect of interference, we need to find the distance between the cuts, δ, that minimizes the performance in

$$\min_{\delta} \max_{\boldsymbol{\alpha}} \min_{Q \in \mathcal{Q}'_\delta} c(Q, \boldsymbol{\alpha}), \tag{13}$$

where $\mathcal{Q}'_\delta$ contains the two cuts at the distance δ from each other. Since for any distance δ we get an upper bound, the minimum with respect to δ in the above equation gives the tightest upper bound with two cuts.

The simulation algorithm for the one-cut case can be modified to apply to the two-cut case, and u can be estimated with (12) where n is the number of times a link crosses either of the cuts, and L is the total length of the two cuts.

4.3 Infinite number of cuts

When the plane is filled with straight vertical cuts, the task of maximizing the number of times a link crosses a cut becomes maximizing the sum of the progresses of the links, that is, finding the maximal weighted independent

set. To clarify this, let n_i be the number of cuts the link e_i crosses. The number is proportional to the progress (i.e., the horizontal length) of the link p_i, and we have $\lfloor p_i/\delta \rfloor \leq n_i \leq \lceil p_i/\delta \rceil$, where δ is the distance between the cuts. Thus, for the proportion n_1/n_2 we get

$$\frac{p_1/\delta - 1}{p_2/\delta + 1} < \frac{n_1}{n_2} < \frac{p_1/\delta + 1}{p_2/\delta - 1} \Leftrightarrow \frac{p_1 - \delta}{p_2 + \delta} < \frac{n_1}{n_2} < \frac{p_1 + \delta}{p_2 - \delta}.$$

So when the number of cuts per unit length goes to infinity, $\delta \to 0$, and $n_1/n_2 = p_1/p_2$. Thus, the contribution of a link is equal to its progress times C, and in order to get the optimal value for u, we need to find the independent set of links with maximal total progress.

The argumentation for the optimality is the same as in the two-cut case. The lines are in equal positions and infinitely long, and thus, by maximizing the sum, we maximize the minimum[4]. Because the set of cuts is still a subset of all the cuts, the result is an upper bound for the sustainable flow. This follows also easily from noting that the same independent set cannot be used repeatedly in consecutive time slots, since the corresponding flow network is not even connected (the capacity of the minimum cut is zero). Actually, we get the maximum total progress that can be achieved in one time slot.

The estimate for u can also be derived from the same interpretation of I^* as in (12). For some small distance δ between the cuts ($n_i \approx p_i/\delta$) and some large width X,

$$u \approx \frac{1}{C\sqrt{\lambda}} \cdot \frac{C \sum p_i/\delta}{LX/\delta} = \frac{\sqrt{\lambda} P}{\lambda LX} \approx \frac{\sqrt{N_R}\, P/R}{N\sqrt{\pi}}, \qquad (14)$$

where $P = \sum_i p_i$ is the total progress of the links in the independent set, and $N \approx \lambda LX$ is the total number of nodes in area LX. The simulation algorithm to be introduced later for this case is again based on the same idea, but modifications arising from the impact of (14) compared to (12) are required.

5. MOVING WINDOW ALGORITHM

In this section, we devise simulation algorithms similar to Retrospective optimization [12] for obtaining an upper bound for the local forwarding capacity in the three cases with different $\mathcal{Q}'$. In these three algorithms, a window separating the links above and below is being moved along the cuts in $\mathcal{Q}'$. A binary tree represents all the possible link combinations in the window area and the value assigned to each leaf shows the maximum size of the (weighted) 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. Since the entering and exiting links are independent, we can combine the on- and off-branches corresponding to a link that has been dropped out of the window and choose the greater values for the new tree. In this way we can recursively find the size of the maximum independent set of links. The algorithms limit in no way the length of the simulation, and when the execution is continued, the result converges without bias towards the true value. Note that the realization of the network can be generated on the fly, simultaneously as the window moves.

[4]Note that the argument for the equality of the cuts holds for two cuts and an infinite number of cuts but not, e.g., for three cuts where the cut in the middle is in different position than the outer ones.

5.1 One cut

With one cut, the problem is to find the maximum independent set of links crossing an arbitrary line in the infinite network. We do this with an algorithm called Moving window algorithm (MWA). Let us now consider a window with the width of two transmission radii R, the height of $3R$, and the line representing the cut going straight through the middle of the window in vertical direction (see Figure 1). Set theoretically the window is represented with a set of nodes W, while the set of links crossing the line in the window area is denoted with L. Considering simulation step i, we have the following definitions (l and r refer to the left and right sides of the network):

$$W_i = W_i^l \cup W_i^r,$$
$$W_i^l = \{v \in V | x(v) \in [x_0, x_0 + R], y(v) \in [y_i, y_i + 3R]\},$$
$$W_i^r = \{v \in V | x(v) \in (x_0 + R, x_0 + 2R], y(v) \in [y_i, y_i + 3R]\},$$
$$L_i = \{e \in E | t(e) \in W_i^l \wedge r(e) \in W_i^r\}.$$

Widening the window would not increase the number of links in L, since no node to the left or to the right of the window is able to form a link with a node that is located on the opposite side of the line. At the same time, all the links with at least one end below the window (e_b) are always independent from the links with at least one end above or at the upper bound of the window ($e_a \notin \mathcal{I}_B(e_b)$). The total length of such two links is always less than $2R$, leaving more than one R between them as a sufficient margin.

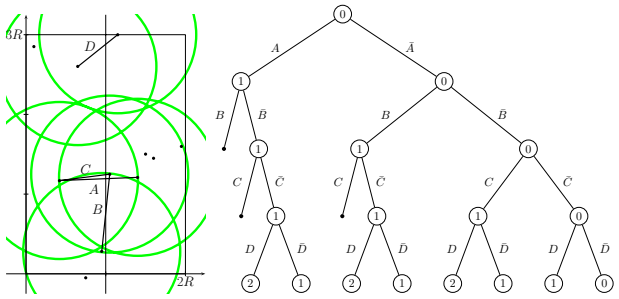

Figure 1: A window with four links crossing the cut and the corresponding binary tree.

All the possible combinations of links in L are represented by a rooted binary tree T. Every level of the tree, i.e., vertices at the same distance from the root, (except the root itself) corresponds to a link in the window, and every edge describes the on-off state of that link in the link combination represented by the vertex incident with the edge and further away from the root. The value assigned to each leaf represents the size of the maximum independent set of links crossing the line so far given the combination of active links in the window.[5] By looking at the tree depicted in Figure 1, we see how each link in the window is either on (A) or

[5]Let i be the current step in the simulation, $G_i = (\bigcup_{j=0}^{i} W_j, \bigcup_{j=0}^{i} L_j)$ and $Q_i = (\bigcup_{j=0}^{i} W_j^l, \bigcup_{j=0}^{i} W_j^r)$. Each leaf, $\xi_i^k = (L_i^{on}, L_i^{off})$, of the binary tree T_i describes whether a link, $e \in L_i = L_i^{on} + L_i^{off}$, is on or off. If $e \in \mathcal{L}_j \cap L_i^{on}(/L_i^{off})$, then $t_j = 1(/0)$ in schedule $\boldsymbol{\alpha}$. The value assigned to a leaf ξ_i^k is thus $\max_{\boldsymbol{\alpha}|\xi_i^k} c(Q_i, \boldsymbol{\alpha})$.

off ($\bar{A}$). The empty vertices in the figure have been added to illustrate the combinations that are impossible due to the interference.

As mentioned earlier, the idea behind the algorithm is to move the window along the line. The algorithm consists of the steps presented in Table 1.

Table 1: Moving window algorithm.

0.	Initialize
1.	Draw step
2.	Drop nodes, remove links, update tree
3.	Add node, create links, update tree
4.	Goto 1.

During the initialization phase, all the variables including W, L, and T are initialized. The first actual step is to draw the location of the next node from the exponential distribution. Since the nodes are located according to a Poisson process, the vertical distance between two consecutive nodes in the path of the window is exponentially distributed with the parameter $2\lambda R$.

Step 2 is to remove all the nodes whose vertical distance from the new node is greater than $3R$ from the window. Accordingly, all the links that the removed nodes are incident with are removed from L. Because these links cannot interfere with the links that become possible when the next node enters the window, they can also be removed from the tree T.

A link in the window corresponds to a level in the tree. Since the link that is being removed from the window cannot interfere with the links that enter the window later in the simulation, it is not necessary to know whether we are in the on- or off-branch corresponding to that link when adding a new link to the bottom of the tree. This means that we can compare the on- and off-branches and choose the maximum of the two in each vertex of the subtree to be the corresponding value (the value of the vertex above) in the updated tree. If the exiting link does not correspond to the top level of the tree, the same procedure is done to all the on-off pairs.

Step 3 is to draw the horizontal location of the new node entering the window from uniform distribution and add the node to the set W. The new node now corresponds to the top of the window. If it is possible for the new node to form links crossing the cut, they are added to L, and also a new level of vertices is added to the bottom of the tree T for each new link. To every leaf of the tree, the off-branch is added without increasing the size of the maximum independent set. The on-branch with the increased value of the maximum independent set is added if the link combination is possible considering the interference.

If the sum of the steps exceeds the simulation length, the simulation ends. Otherwise, the distance to the next new node is drawn.

5.2 Two cuts

When the number of cuts is two, the algorithm works almost as it is. The second cut needs to be added, and the window must be made broader to cover both cuts. If the distance between the cuts is greater than $2R$, the cuts become independent, and two separate windows could be used. This corresponds to two parallel one-cut simulations. Additionally in Step 3, the value in the tree increases by two if the corresponding link crosses both cuts. The two-cut version of the algorithm is referred to as MWA_2.

5.3 Infinite number of cuts

For the case with infinite number of straight cuts, the algorithm needs to be modified a bit, although the basic working principle stays the same. If we limit the width of the window for simulation purposes, we lose the symmetry since the cuts near the border are in unequal position. An infinite width is infeasible, but the harmful border effects in the horizontal direction can be diminished by connecting the opposite sides of the window together to form a tube with the intention to extrapolate the value of u for an infinitely wide tube/window. The perimeter of the tube still needs to be large enough to fit several consecutive links. This, combined with the fact that none of the links can now be discarded straight away since they all cross a cut, places limitations on the simulation parameters and increase the memory requirements compared to the previous methods.

To overcome some of the restraints, the new algorithm MWA_∞ drops the links straight after they stop interfering with the links that enter the window in the near future. Although only the necessary links are kept in the binary tree, the size of the tree may still grow rapidly and become infeasible[6] due to the stochastic nature of the processes. For this kind of situation, we place a restriction for the size of the tree. When the number of nodes in the tree grows larger than this value, we start to remove links from the tree. By removing the shortest links first, we make sure that the error made is relatively small (less than 0.1% in the tests).

6. RESULTS

Figures 2 and 7 show $u(N_R)$ for MWA with one and two cuts. The corresponding data are given in Table 2. In Figure 2, the solid lines are fitted third degree polynomials, while the error bars represent the 90% confidence intervals. Figure 3 shows how the dimensionless mean progress behaves as the distance between the cuts is changed in MWA_2. The task is to find δ that minimizes the maximal u, i.e., corresponds to the saddle point of the surface in the Figure 3. The dimensionless mean progress, $u(N_R)$, is minimized when $\delta \approx 0.925$, which is the value used for Figure 2.

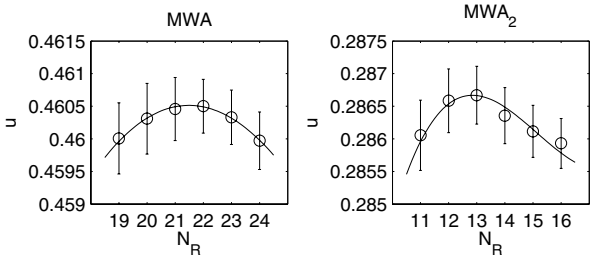

Figure 2: $u(N_R)$ for MWA (left) and MWA_2 (right).

Since the interference model favors links on top of each other in towerlike formations, u in MWA_∞ starts to stabilize only when the tube is wide enough to fit several consecutive links, as can be seen from Figure 4. The maxima appear

[6]The 2GB memory limitation of 32-bit architecture

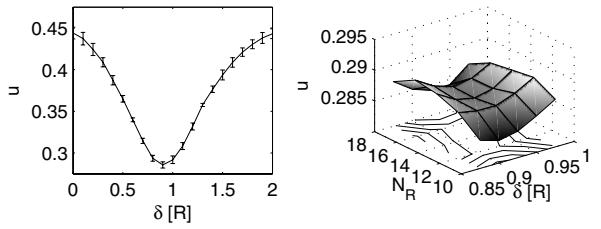

Figure 3: $u(\delta)$ **for** $N_R = 12$ **and** $u(N_R, \delta)$**.**

when the tube is wide enough to hold full length links and the margins between them, that is, when the perimeter of the tube x $[R]$ is even. Since the behavior of u is quite regular, a damped oscillating curve

$$u(x) = Ae^{-Bx}\cos(Cx + D) + E \tag{15}$$

has been fitted to the data. The constant term of the curve, E, gives the actual value for $u(N_R)$.

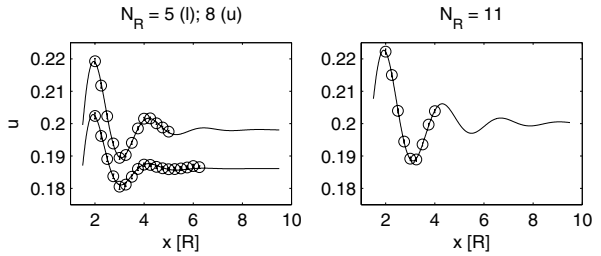

Figure 4: $u(x)$ **for different values of** N_R **when (15) has been fitted to the data.**

When $N_R = 11$, we are only able to simulate data points around the first minimum. This is not enough to fix the positions of the minima and maxima, which partly explains why the frequency and phase terms of the curve clearly differ from those of other curves in Figure 5. Based on Figure 5, that represents the parameters of (15) as a function of N_R, we replace the individual constant amplitude and damping terms with linear terms. From the single and two cut versions we also know that the constant term can be well approximated with a polynomial of third degree near the optimum. The fact about the appearance of minima and maxima at regular values of x means that the frequency and phase terms should be constant (and we keep them such), but because of the randomness and the distance between the data points, as well as the scarcity of data with $N_R = 11$, some deviation is visible. Thus, we have the surface

$$\begin{aligned} u(N_R, x) =& (A_1 N_R + A_2)e^{-(B_1 N_R + B_2)x}\cos(Cx + D) \\ &+ (E_1 {N_R}^3 + E_2 {N_R}^2 + E_3 N_R + E_4) \end{aligned} \tag{16}$$

to be fitted to the data. The resulting parameters from this fit are also plotted in Figure 5 (dashed lines). All the available data points were used, and the resulting surface is depicted in Figure 6.

6.1 Comparison

Figure 7, that depicts u as a function of N_R, and Table 2 summarize the results from each of the used methods. The Moving window algorithm with one cut (MWA) gives

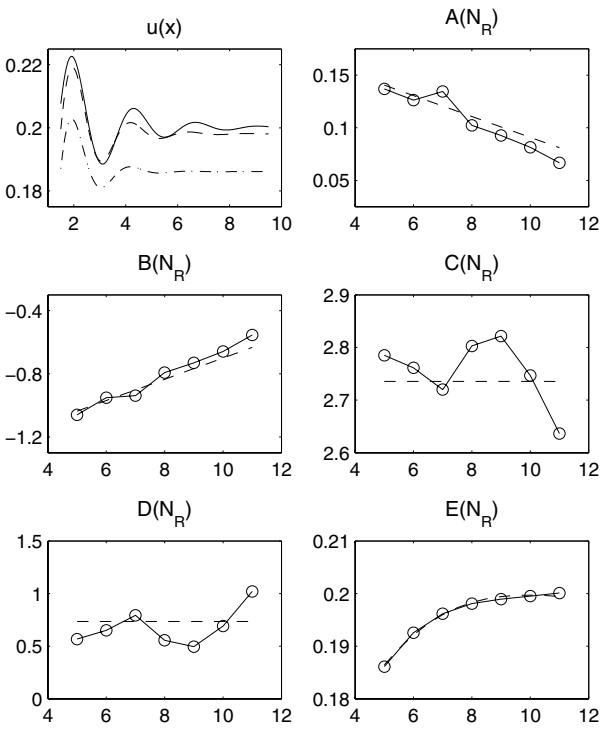

Figure 5: $u(x)$ **from Fig. 4 and the corresponding parameters. The dashed lines give the same values for (16).**

a considerably high upper bound for the performance since it does not consider the interference in horizontal direction. The same algorithm with two (MWA_2) or infinite number of cuts (MWA_∞) gives more moderate values. The dimensionless mean progress given by the MWA with infinite number of cuts is the maximal forwarding capacity over one time slot. The same performance cannot be achieved in consecutive time slots, and thus, it still gives an upper bound for the maximal achievable flow.

The Opportunistic forwarding method (OF) [1] represents an actual forwarding method and hence gives a distinct lower bound for the real maximal forwarding capacity. OF also demonstrates what can be achieved with a random access method using local coordination. To improve viable forwarding methods, one would thus have to consider more global coordination or more efficient access methods. In OF a node with packets broadcasts with a constant probability p to all forward neighbors, and from the forward neighbors able to receive the packet, the one with the greatest progress is chosen to be the next hop. Thus, the receiving node $j = \arg\max\, d_{ij}\mathbf{1}_{R_j}$, where $\mathbf{1}_{R_j} = 1$ if neighbor j receives the transmission, i.e., there is no collision, and zero otherwise. Opportunistic forwarding is a variation of Extremely Opportunistic Routing (ExOR) [3], but it has been modified mainly to avoid duplicate packets. When no price has been placed for the procedure that chooses the receiver after the transmission, it is generally the most efficient of the proposed forwarding methods. The results for OF have been obtained from simulations using an improved version of the simulation model in [1].

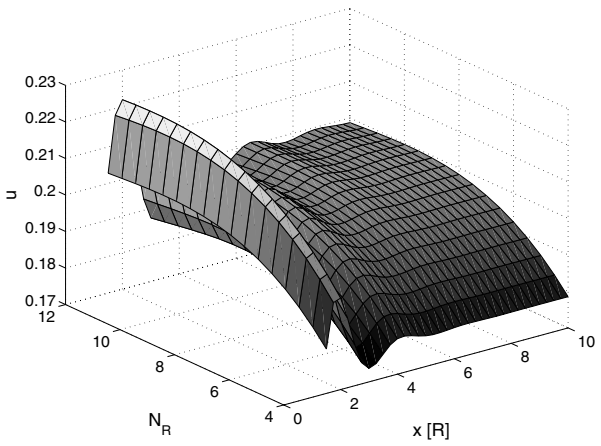

Figure 6: $u(N_R, x)$ when (16) is fitted to the data.

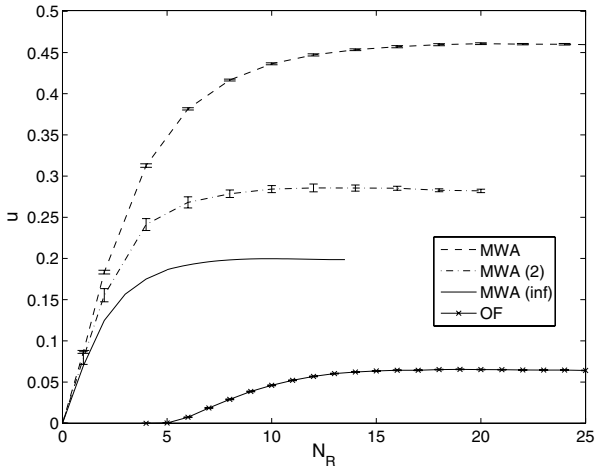

Figure 7: $u(N_R)$ for various methods.

7. CONCLUSIONS

We have used a network model where the separation of spatial scales occurs and studied the microscopic level forwarding capacity. The maximal forwarding capacity is a medium-specific constant that can be divided between different directions, e.g., via time sharing. Thus, with a minimal loss a generality, it is enough to consider relay traffic traversing in a single direction. Because of the computational complexity of the maximum flow problem in wireless multihop networks, the task is approached through simulations. With the devised algorithm, MWA, it is possible to advance to a situation that corresponds to the maximum capacity over one time slot. The tightest upper bound for the maximum forwarding capacity is about three times the highest dimensionless mean progress achieved via feasible random access forwarding method that uses local coordination.

Since the upper bound is still not tight, but actually the one-slot maximum, the real room for improvement is yet unknown. One topic for future work is to investigate how far the real forwarding capacity is from this. Another could be to test the algorithm with other types of interference models.

Table 2: The maximal dimensionless mean progress, u, and the corresponding N_R for various methods discussed in this paper.

	u^*	N_R^*
MWA	0.461	21.6
MWA_2	0.287	12.5
MWA_∞	0.199	9.9
OF	0.065	19.0

8. REFERENCES

[1] O. Apilo, P. Lassila, and J. Virtamo. Performance of local forwarding methods for geographic routing in large ad hoc networks. In *Proceedings of The Fifth Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net 2006)*, 2006.

[2] 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.

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

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

[5] T. Coenen, M. de Graaf, and R.J. Boucherie. An upper bound on multi-hop wireless network performance. In *International Teletraffic Congress*, volume 4516, pages 335–347, 2007.

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

[8] 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.

[9] 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.

[10] L. Popa, A. Rostamizadeh, R. Karp, C. Papadimitriou, and I. Stoica. Balancing traffic load in wireless networks with curveball routing. In *Proceedings of ACM MobiHoc*, pages 170–179, 2007.

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

[12] J.T. Virtamo. A model of reservation systems. *IEEE Transactions on Communications*, 40(1):109–118, 1992.