Communication over a hybrid ad hoc wireless network

M. Palanques, Y. Bar-ness and A. Laufer

Gràcies / Thanks

Gràcies al Iaio, per donar-me en herència la seva curiositat, les seves ganes d'aprendre i el seu esperit cientific. Em conformo si influeixo a la vida de la meitat de la gent a la que has influït tú.

Gràcies a la resta de la meva familia, en especial als papis i a l'Enano, perque heu estat la meva companyia inseparable desde el principi, inclús quan he estat lluny, perque sou la meva casa i sempre esteu amb mi.

Gràcies al Javi perque també està sempre al meu costat, per esperar sempre el millor de mi i per saber com motivar-me. Saps que la meva vida i la meva carrera no serien així si no fos per tú i espero que segueixis al meu costat, perque treus el millor de mi. How wonderful life is, while you're in the world…

Gracias a Rebe por su apoyo incondicional, por las horas de charlas hasta las tantas y por una lista inacabable de detalles. Sin duda, eres todo lo que se puede pedir de una amiga y mucho más.

Thanks to Professor Bar-ness for giving me the chance of living the incredible experience of living in Newark and working side by side with outstanding people.

Gràcies a Jordi Perez per acceptar dirigir el meu projecte i per animar-me a acabar-lo aviat.

Thanks to Erika and Carlos, for showing me "the real New Jersey", for taking care of me so many times, for letting me be part of your awesome lifes, but specially for loving me and making me feel like home when I was thousands of miles away. I'm so glad Carlos didn't have a lighter that day… You're both extraordinary! I miss you like crazy and hope I'll see you soon. Thanks to Amanda, JD, Pax, Don, Steve… I think about all of you.

Thanks to Pete and Matt for giving me a place where to stay, making me feel comfortable, driving me to the Shoprite a thousand times, letting me eat your girl scout cookies and much more... You know you've got a home in Barcelona whenever you need it.

Gràcies a l'Alex per haver aguantat totes les meves preguntes i aconsellar-me en tantes coses, i haver-me tret de passeig quan necessitava companyia. Tens una dona excepcional! Gràcies a ella també.

Gràcies a Gresca, per haver-me permés coneixer gent increíble, per haver evitat que passés massa hores a la biblioteca i haver-me ensenyat lliçons que no es poden obtenir a un aula. Gràcies a la Senda per no ser "una noia de gresca". Gracias a Nachete por liarme para entrar y a todos los que han matado las horas en el despacho o el poli conmigo, los que han cambiado una bombilla y a cualquiera que haya derramado una gota de sudor a mi lado. We made it possible!

Index of Contents

Index of figures

Index of Tables

Section I. Introduction

The main advantage of wireless networks is the fact that connections don't exist over a physical support such as a cable, but they exist over what we name a shared media, i. e. air. This means that one can theoretically connect any pair of nodes without a cable constraint and also that the communication channel (air) might be shared with other pairs of communicating nodes, and confers high flexibility to the network.

Though one can try to have a large set of pairs of nodes transmitting at the same time this way, one must also take into account that all transmissions cause interference on the other ones and that a node is just able to receive or transmit a single transmission at the same time. It is also remarkable that although one might try to establish a direct communication between any pair of nodes, chances are that the channel has a bad quality between a random pair of nodes and that if they need to communicate, they will have to use relay nodes. Last, one also usually needs a connection between the wireless network and a wired network (such as the internet or the plain telephone network). In this case, there will be one or several nodes connected to the wired network and the remaining of the wireless nodes that need to reach the wired network will have to use those nodes as relays, which creates hot spots in the nodes that have access to the wired network. For all those reasons, having a scheduling that organizes all the simultaneous transmissions assuring that one has a reasonable amount of simultaneous transmissions but there are no collisions at the same time turns to be a critical point in wireless networks and it can lead to significant variations on the network performance.

The simplest and most common wireless network approach is the wireless network with infrastructure and a hierarchical structure, where one has an access point (usually connected to a wired network) that manages the transmissions of the wireless nodes and the access of them to the wired network. In these networks, every transmission uses the access point as relay node, meaning that the source transmits directly to the access point and this one transmits directly to the destination. We call cluster a set of wireless nodes connected to a certain access point. The main drawback of such a network is that every node needs to be connected to the access point. This scheme is, for instance, the one used in mobile communications.

In Figure 1, we can see the scheme of a hierarchical wireless network with infrastructure, where nodes are labeled with numbers. Every access point (nodes BS1, BS2 and BS3) manages the wireless nodes connected to it. We will use this labeling for nodes and access points in all our work.

For instance BS3 manages the transmissions where (1), (2) and (7) are involved. When node (7) wants to communicate with node (5), it transmits to BS3, which contacts the access point associated to node (5), i. e. BS2, and transmits towards it, so BS2 can deliver the information to the receiver (5). Note that all nodes need to communicate using the access point, thus nodes connected to a same access point cannot transmit or receive at the same time without colliding. In Figure 1, nodes (3), (4) and (5), for instance, cannot communicate during the same time slot.

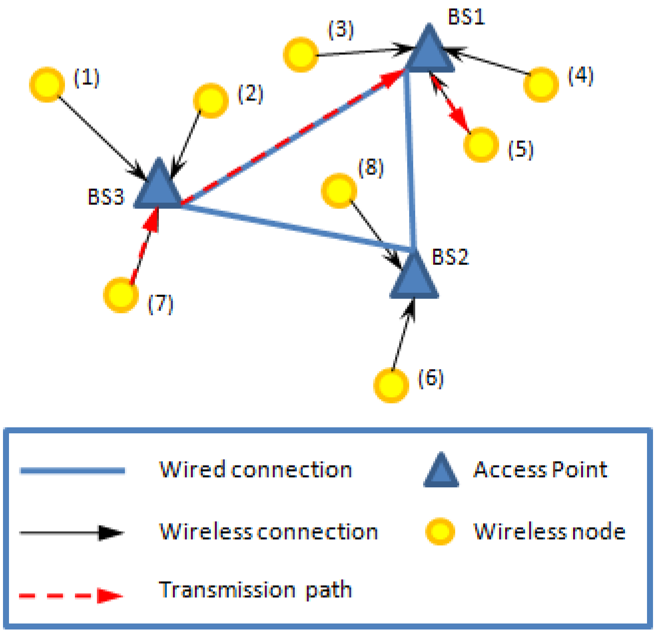

Fig. 1: Hierarchical network scheme

However, sometimes a set of wireless nodes may need to communicate without the support of an access point, this is, without hierarchy. In an ad hoc network, as they are called, the message reaches the destination by forming a path that uses other wireless nodes as relay. These networks have the drawback of a more complicated scheduling, due to the lack of a coordinator. On the contrary, they have the advantage of being independent and more flexible, meaning that the network can be created by any set of nodes, without need of an infrastructure (access point) and any node may enter or leave the network anytime if needed.

A node just needs to have a good connection to some other node in the network to be part of it, instead of concretely needing a good connection to the access point.

Figure 2 shows an ad hoc network scheme, where some of the nodes are connected to the internet. Note that they are regular wireless nodes, but with a connection to an independent wired network that doesn't help in the transmission. A message transmitted from node (1) to node (6) crosses a path of wireless nodes that is chosen attending to some scheduling criteria. If a node wants to communicate with an external network, it will transmit to a wireless node that is connected to the intended destination network, which are nodes (7) and (10) in the example, and this node will transmit towards the external network.

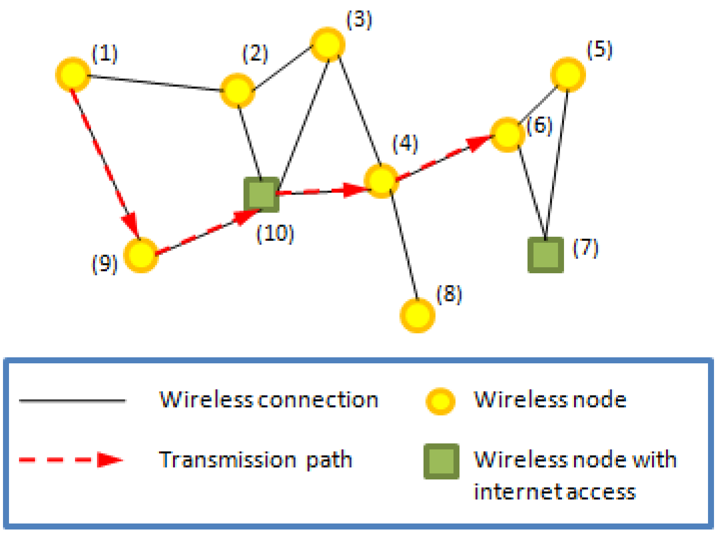

Fig. 2: Ad hoc network scheme

Our work focuses on a network that is halfway through those two models: a hybrid ad hoc network (an ad hoc network with infrastructure support). This is, an ad hoc network where some of the nodes are fixed and connected between them by wired links. In the hybrid network, nodes might communicate across the infrastructure network or through a path of relay nodes and even use the infrastructure network after using relay nodes to reach an access point (which is called a hybrid path).

Figure 3 illustrates this type of network and a hybrid path that connects node (9) to node (4). The path performs an ad hoc connection from (9) to (7) and then a cellular connection using the infrastructure from (7) to (4). Note that in this example it isn't possible to have a pure cellular path between those two nodes, but it would be possible to reach node (4) from node (9) through the ad hoc path formed by the sequence (9) – (8) – (5) – (4). This path would be shorter, but this doesn't necessarily imply that it would be faster, because the wired link usually has a higher capacity.

On the other hand, most of the work done on ad hoc, and more generally on wireless network's throughput and capacity performance, assume a transmission model where the connections strength decreases with the distance (geometric model). O. Dousse, P. Thiran and M. Hasler's work in [1] and P. Gupta and P. R. Kumar's [2] are good examples of such networks. The classical geometric connection approach fits an environment where the decrease of the strength of a signal due to the distance has an effect that is stronger than the effect of the noise or the reflections. However, in a highly scattered environment, the strength is more likely to be random, and this is the case that we are interested in.

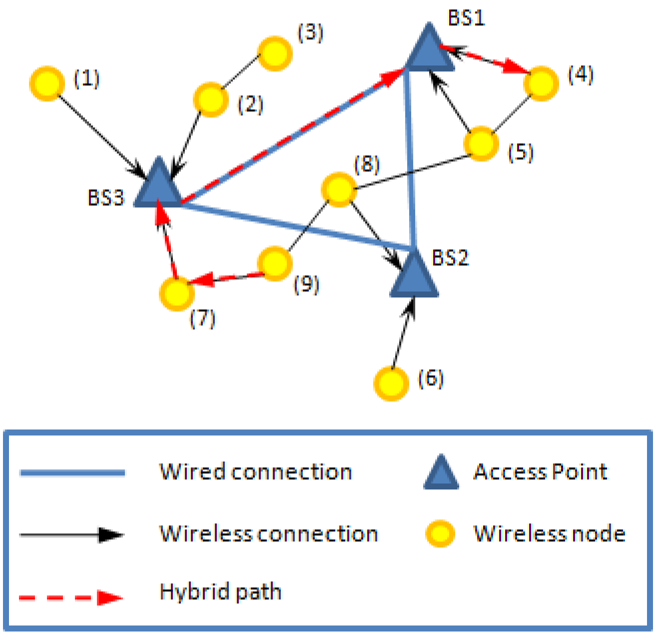

Fig. 3: Hybrid network scheme

The aim of this work is to develop an analytic model to evaluate the throughput that a hybrid ad hoc network with random connections can achieve with help of an infrastructure network, with the objective of publishing the results in a paper in the IEEE Wireless Communications Magazine.

R. Gowaikar, B. Hochwald and B. Hassibi [3] introduced and studied a pure ad hoc network with random connections. Our work focuses on the throughput improvement introduced by the addition of infrastructure nodes to their scheme, the minimum infrastructure requirements, and the scaling of the throughput as a function of the infrastructure size. We also introduce a different scheduling scheme since the one in [3] is very suboptimal.

In Section II, we summarize the work in [3], analyze its performance and make some comments on the suitability of its characteristics to a hybrid network model. In Section III, we define the models of our network and signal, and in Sections IV, V and VI we introduce three approaches to analyze the throughput when adding access points to the network: first in Section IV we use the same analysis as in [3] and find out that the addition of infrastructure nodes basically increases the number of simultaneous transmissions that one can schedule, then in Section V we force the paths to use the infrastructure and improve the throughput and in Section VI, by ruling out the use of vertex-disjoint paths, we re-analyze the overall performance of the system, resulting in a higher improvement in the achievable throughput. In Section VII, we introduce a more accurate definition of the network behavior for the models in Sections V and VI. And finally we study the results obtained for some concrete distributions of the connection strengths and numerical values in Section VIII, comparing between them and in Section IX we extract conclusions of our work and give some guidelines for future research.

Section II. Previous work - Model and results

In [3], R. Gowaikar, B. Hochwald and B. Hassibi study the achievable throughput of a pure ad hoc network with random connection strengths that schedules the transmissions finding vertex-disjoint paths, i. e. paths that do not share any node during all the time needed to cross them. To that purpose, authors use the algorithm and results introduced in [4]. Note that, although [3] is the main basis for our work, they differ in two main aspects: (i) the network studied in [3] has no infrastructure and (ii) the analysis and results are subject to the use of the algorithm in [4] and the vertex-disjoint paths.

a) Model

The model in [3] assumes a network with *n* nodes in which the connection strengths between them are drawn independently and identically from an arbitrary distribution with pdf $f_n(\gamma)$, cumulative distribution $F_n(\gamma)$ and complementary cumulative distribution $Q_n(\gamma)=1-F_n(\gamma)$. In a classical ad hoc network where connections strengths depend on the geometrical layout of the nodes, a node may directly communicate with another if they are close enough. In the proposed random model, however, this geometric notion doesn't exist. Instead, the paper defines the concept of *good connection* as a connection between two nodes *i* and *j* with strength $\gamma_{ij}=|h_{ij}^2|$ greater than a chosen threshold β_n, and they call the connectivity $p=P(\gamma_{ij}\geqslant\beta_n)=Q_n(\beta_n)$, *i. e.* the probability that a certain connection is *good*, which generally depends on the number of nodes in the network *n* and on the chosen threshold β_n. Thus, this scheme consist of *n* nodes and $\binom{n}{2}$ random variables, which are the strengths of the connections, that are assumed not to change in time. In the following, we will indistinctly refer to *p* and $Q_n(\beta_n)$.

Note that the concept of *good connection* just considers the channel response, that doesn't take into account the interference effects. If *k* nodes (i_1, i_2 ... i_k) are transmitting signals of power *P* at the same time, and the channel has additive white gaussian noise with variance

σ^2 , the signal-to-interference-plus-noise ratio (SINR) between a transmitting node i_1 and the receiving node j is:

$$\rho_j = \frac{P \cdot \gamma_{i_1, j}}{\sigma^2 + P \cdot \sum_{l=2}^{k} \gamma_{i_l, j}} \qquad (1)$$

Since the threshold for a *good connection* doesn't consider the existence of interference, there's a chance that a transmission over a *good connection* isn't possible. That is transmission is not possible whenever $\rho_j < \rho_0$, where ρ_0 is the smallest SINR needed to accomplish a transmission between two nodes, which is shown by the authors of [3] to be:

$$\rho_0 = \frac{a_n \cdot \beta_n}{\frac{\sigma^2}{P} + (k-1)\mu_\gamma} \qquad (2)$$

where a_n is a factor smaller than *1*, and μ_γ is the mean of γ.

Thus, the transmission rate[1] of any link is $log(1+\rho_j)$, which is higher or equal to $log(1+\rho_0)$. In order to simplify the analysis, authors of [3] use $log(1+\rho_0)$ (the lower bound), instead of $log(1+\rho_j)$, and so will we do.

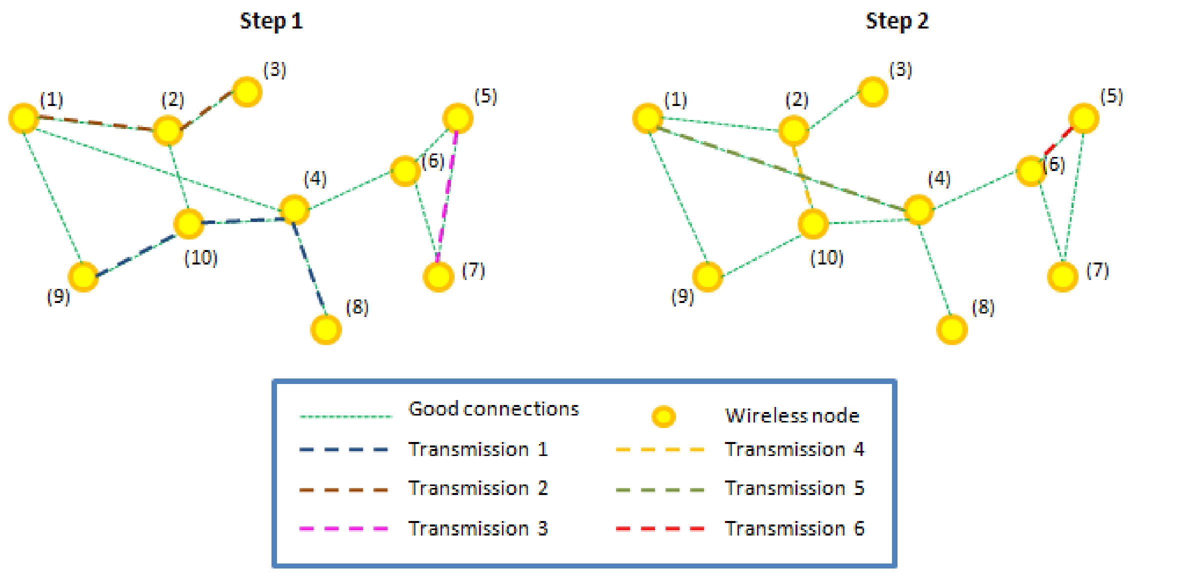

Fig. 4: Use of vertex-disjoint paths in a pure ad hoc network

1 The transmission rate is normalized per herz (i. e. spectral efficiency).

The transmission of every message is done by using a sequence of relay nodes, i. e. performing *h* hops, and therefore needs *h* time slots to be completed. Figure 4 shows how the scheduling works. First, a set of pairs source-destination with the related vertex-disjoint paths is found. Those paths are simultaneously used during step 1, which lasts 3 time slots (the longest path's duration). After that, a new set of pairs source-destination and paths is found and it is used during step 2. The duration of a step is always the duration of the longest path performed during it, which is upper bounded by the network diameter[2].

Assuming that the scheduling allows us to schedule *k* simultaneous transmissions during *h* time slots with some probability of error in the transmission of every message *(ε)*, the achievable throughput is defined as:

$$T = (1-\epsilon)\frac{k}{h}\log(1+\rho_0) \tag{3}$$

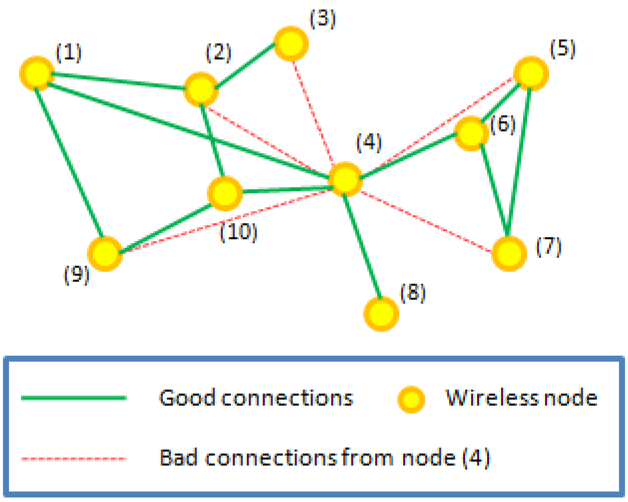

Fig. 5: Random graph illustration, including all good connections and bad connections from node (4)

Consider our network as a graph *G(n,p)* with *n* vertex and edges, where each connection or edge exists with independent and identically distributed probability *p*. Since the connection strengths are constant along time, the graph doesn't change either. An example of graph is drawn in Figure 5. Due to the density of *bad connections*, the figure only contains the ones that concern node (4), which has *good connections* to (1), (6), (8) and (10) and b*ad connections* to (2), (3), (5), (7) and (9). This means that (4) can only communicate directly

2 The diameter of the network is the highest among the minimum distances, in terms of hops, between any pair of nodes in the network.

with (1), (6), (8) and (10) , but it needs to use relay nodes if it wants to reach any other node. Each of those connections is *good* or *bad* with a certain probability *p* that is independent of the *goodness* of the other ones.

Graphs like this have some known properties[3] that lead to restrictions on the transmission and the network. It is known that, for such graph, when $p>\frac{\log n}{n}$, the probability that it has no isolated vertices (nodes not connected to others) goes to one rapidly. Thus this condition is necessarily assumed. With this condition, the diameter of the graph (the maximum distance between any pair of vertices of the graph) behaves like $\frac{\log n}{\log(np)}$ and so does the average distance between two nodes. Since the number of hops that a message needs to perform to reach its intended destination is at most the diameter, this is bounded by

$h=\frac{\log n}{\log np}$ [4].

This scheme of [3] also requires the condition that all relay nodes used in a certain time slot are different and that a certain relay node cannot be transmitting and receiving a message at the same time, which they call the non-colliding condition. To assure that this condition is accomplished, the authors find the paths from every source to its destination by using the algorithm introduced by A. Z. Broder, A. M. Frieze, S. Suent and E. Upfals in [4], the result of which is a set of *k* vertex-disjoint paths in the graph formed by nodes and their related good links *G(n,p)*. Note that the vertex-disjoint condition[5] is stronger than the network needs, and thus the use of this algorithm is sub-optimal.

The work in [4] shows that using the non-colliding paths condition put forth by the algorithm, one can schedule at most

---

3 One can find these properties' demonstrations in [5]-[8].

4 One can find a proof of this, for instance in Section *10.2 The Diameter of* G_p of [5].

5 The paths found by the algorithm in [4] are vertex-disjoint, which means that none of the paths that are crossed simultaneously during the interval of *h* hops share any node. This implies that every node is used at most by just one path every *h* hops. However, the system is actually able to use every node every time slot, thus the vertex-disjoint condition is much stronger than our needs.

$$k \leqslant \alpha n \frac{\log np}{\log n} \tag{4}$$

simultaneous transmissions for some *α>0* and, since every node can only be used once every *h* time slots, the number of hops *h* performed by every message is asymptotically

$$h = \frac{n}{k} = \frac{\log n}{\alpha \log np} \tag{5}$$

Note that the maximum number of simultaneous transmissions that one can schedule is a function of *n* and *p*, which depends on the size of the network and the chosen threshold. Authors of [3] also refer to *k* as $k_n(\beta_n)$ to emphasize the dependence on those two parameters. We will name it *k* for simplicity.

Finally, the authors of [3] find an upper bound to the probability of error in the transmission of a message from source to destination, which is given by:

$$\epsilon_n \leqslant \frac{\log n}{\alpha \log np} \frac{\sigma_\gamma^2}{(k-1)\left(\frac{P\beta_n - \rho_0 \sigma^2}{(k-1)P\rho_0} - \mu_\gamma\right)^2} \tag{6}$$

where μ_γ and σ_γ^2 are respectively the mean and the variance of the strength *γ* of the links. The authors require ε_n to go to zero as $n \to \infty$, which can by satisfied by using the adequate *k*.

b) Main result

The results of the paper are summarized in a Theorem that states that:

A network with *n* nodes and a given connection strength threshold β_n such that the probability of a link exceeding β_n behaves like $Q_n(\beta_n) = \frac{\log n + w_n}{n}$ with w_n going to infinity when *n* also does, the following throughput is achievable for a positive constant *α*:

$$T = (1-\epsilon_n)\alpha\, k \frac{\log(n\,Q_n(\beta_n))}{\log n} \times \log\left(1 + \frac{a_n \cdot \beta_n}{\frac{\sigma^2}{P} + (k-1)\,\mu_\gamma}\right) \tag{7}$$

for any $0<a_n<1$ and any k that satisfies the following conditions:

1) $k \leqslant \alpha\, n \frac{\log(nQ_n(\beta_n))}{\log n}$
   (4 bis)
2) $\epsilon_n \leqslant \frac{a_n^2}{\alpha(1-a_n)^2} \frac{(k-1)\cdot\sigma_\gamma^2}{(\frac{\sigma^2}{P}+(k-1)\,\mu_\gamma)^2} \frac{\log n}{\log(nQ_n(\beta_n))} \rightarrow 0$
   (8)

The expression in (7) is obtained by the substitution of (2) and (5) in (3).

The authors also include a corollary stating that, whenever $\frac{\sigma^2}{P} - \mu_\gamma > 0$ the throughput is maximized by taking k as large as possible.

Note that the theorem states an expression for some achievable throughput, not a bound. Also take in consideration that the goodness of this result remains dependent on p, thus dependent on the distribution $f_n(\gamma)$.

c) Performance for concrete distributions

The second part of the paper [3] studies the performance of this network compared to the classical geometric network, for some different distributions $f_n(\gamma)$. As in a geometric model, the throughput of this network is usually interference limited, and therefore densities that lead to small interference achieve a better performance.

Authors of [3] consider the distribution of a shadow fading model, which corresponds to a situation with strong fading; the distribution that results from the predominance of a decay law and last a distribution that has a mean and variance that are independent of the network size n. The first two models fit certain environment conditions, while the last one is a mathematical model that can correspond to many different situations.

a) *Shadow Fading Model*

A strong shadow faded model has a pdf like $f_n(\gamma)=(1-p_n)\delta(\gamma)+p_n\delta(\gamma-1)$. A natural choice for β_n is 1, such that $Q_n(\beta_n)=p=p_n$. The throughput is almost linear in n when $p\to 0$ and almost constant otherwise. Thus it is optimal to chose p as small as possible. Since one needs $p\geqslant\frac{\log n}{n}$ to assure that the network doesn't have isolated nodes, one will choose $p=\frac{\log n}{n}$. Note that this result states that having the minimum connection probability we can achieve the best throughput. This means that a network can be under-connected if $Q_n(\beta_n)<\frac{\log n}{n}$, but also over-connected if $Q_n(\beta_n)>\frac{\log n}{n}$.

b) *Density Obtained From a Decay Law*

If the decay law of the network has the form $g(r)=\frac{1}{r^m}$, it was shown that one can achieve an almost linear throughput behavior whenever $m\geq 2$ which substantially differs from previous results obtained for structured deterministic model with the same decay law that stated a scaling like $O(\sqrt{(n)})$ and $O(\sqrt{(n/\log n)})$. The improvement is caused by the fact that in the random model, nodes communicate across good links rather than across the shortest ones and therefore the number of hops is reduced from $\sqrt{n}$ to *log n* due to the fact that the interference is drowning a smaller number of nodes and that the distance between any two nodes is decreased to *log n.* These advantages come from the suppression of geometric constraints.

c) *Distribution With Constant Mean and Variance*

In the case that μ_γ and σ_γ^2 are both independent of n, the throughput is always maximized by choosing $k\to\infty$:

$$T=\left(1-\frac{a^2\sigma_\gamma^2}{\alpha^2(1-a)^2\mu_\gamma^2}\frac{\log^2 n}{\log^2(np)}\frac{1}{n}\right)\times\frac{a\alpha}{\mu_\gamma}\frac{\beta_n\log np}{\log n} \tag{9}$$

and we need to choose β_n such that it maximizes $\beta_n \log np$ while $p = \frac{\log n + w_n}{n}$ (recall that $p = Q(\beta_n)$ depends on β_n).

For further study, the paper [3] also analyzes the exponential density, the heavy-tail distribution and the lognormal fading as examples of distributions that fit this model, concluding that the throughput scaling is worse than the one obtained in the previous distributions.

Besides the general result of an achievable throughput, it is interesting to emphasize that the paper [3] states that there is an optimal amount of fading (i. e. an optimal connectivity), that leads to optimal throughput. Any less or any more connectivity will degrade the performance (this last case happens, for instance, with the shadow fading model).

It is also remarkable that the algorithm used to find the paths is clearly sub-optimal. The scheduling works so that the algorithm finds *k* vertex-disjoint paths[6] between *k* nodes pairs and the messages are transmitted over those paths during *h* time slots. This means that any node is going to be used at most twice (one time slot receiving and another one transmitting) every *h* time slots, when they could actually be used every time slot, either transmitting or receiving. This limitation has a worse effect in a network with infrastructure nodes such like the network that we will analyze, since those nodes are indeed capable of handling many simultaneous transmissions in their wired link.

6 Recall that the vertex-disjoint paths don't share any node in all their duration, rather than using different nodes in every time slot.

Section III. Proposed Hybrid Ad hoc Network

As mentioned in Section I, the contents of [3] will be the basis of our work. The network scheme is obtained as a modification of the one in [3] by adding an infrastructure, which consists of a set of nodes that are connected between them by wires but that can also communicate in a wireless mode with the rest of the nodes.

Thus, our proposed network consists of *n* wireless nodes and *m* infrastructure nodes (access points). The access points are strongly connected with high capacity links, and thus we will consider that the probability of any pair of access points to be connected is 1. The wireless connections between nodes and also between any pair node/access point are randomly independent and identically distributed as in the previous work, with pdf $f_n(\gamma)$, cumulative distribution $F_n(\gamma)$ and complementary cumulative distribution $Q_n(\gamma)=1-F_n(\gamma)$. The total number of random variables is thus $\binom{n+m}{2}-\binom{m}{2}$ [7].

a) Probability of good link

In the proposed network scheme, the overall probability of having a good link has to consider all the links in the graph, a total number of $\binom{n+m}{2}$ random variables. If we consider that the links between infrastructure nodes (wired links) are good with probability 1 and the remaining ones (wireless links) are good with probability $p=Q_n(\beta_n)$, the overall probability of having a good link *q* is:

$$q=\frac{\binom{m}{2}}{\binom{n+m}{2}}1+\frac{\left(\binom{n+m}{2}-\binom{m}{2}\right)}{\binom{n+m}{2}}p=p+\frac{\binom{m}{2}}{\binom{n+m}{2}}(1-p) \tag{10}$$

7 The total number of links is $\binom{n+m}{2}$ but the strengths of the $\binom{m}{2}$ wired links are deterministic and ideal.

It is easy to see that the connectivity q is always greater than p, which is the probability of having a good link in a pure ad hoc network.

On the other hand, and since the number of nodes in the new network is $(n+m)$, the new lower bound for the probability of having a good link in order to ensure that our network is connected is

$$q > \frac{\log(n+m)}{n+m} \tag{11}$$

Since the denominator grows faster than the numerator, this condition is always less restrictive than the one found for the pure ad hoc network.

Note that the condition (11) is applied to q, which is larger than p by the existence of the wired links as showed in (10), and thus we have a double improvement. However, in order to simplify the calculations, we will require

$$p > \frac{\log n}{n} \tag{12}$$

to be accomplished in our study.

b) Scheduling

As mentioned in Section I, scheduling is a crucial issue in a wireless network, and it becomes even more important in a network where traffic isn't managed by an infrastructure. The aim of scheduling is to optimize the usage of resources allowing the nodes to be operative as efficient as possible.

Since in a wireless network all messages are transmitted over the same channel, the interference has a significant effect on the performance. Also recall that we assume nodes not to be capable of receiving two transmissions at the same time. If two nodes are transmitting towards the same receiver at a time, both messages will be lost, which has a strongly negative impact on the throughput. The chosen scheduling will have to establish a criteria that allows nodes to transmit knowing that the message won't collide with another one.

The design of the scheduling establishes a bound to the number of simultaneous transmissions that the network is able to perform in a time slot. Given this bound, one will have to choose a number of simultaneous transmissions that maximizes the throughput but always maintaining a compromise between this and the amount of interference. This means that, in some cases, it will be desirable in terms of throughput to decrement the number of simultaneous transmissions.

The problematic of scheduling in an ad hoc network, and more generally in a wireless network, has been object of wide study and isn't the main objective of this work. The scheduling's achievements have a big influence in the results here exposed, but in most cases we will just assume that one knows a scheduling which is capable of accomplishing the required conditions, without analyzing its operation.

In the following, we will consider different approaches for scheduling schemes for the hybrid network, being interested in determining the behavior of the variables that have an influence on the throughput according to expression (3) and the throughput itself, rather than in the operation of those schemes. For that purpose, we will use the definitions in [3] regarding the concept of *good connection*, the SINR in (1), the SINR threshold in (2) and the fraction of lost messages ε in (8).

Section IV. First scheduling approach for hybrid network: using vertex-disjoint paths in the hybrid network model

In our first approach, we will repeat the analysis performed in [3] considering that the addition of *m* infrastructure nodes has two effects in the network: (i) the size of the network increases from *n* to *(n+m)* and (ii) the overall connectivity *q* of the network is increased by the fact that the infrastructure nodes are connected between them with probability 1 in comparison to the previous connectivity *p*. We won't modify any of the criteria established by [3], thus will still use the vertex-disjoint path finding algorithm in [4], although this is obviously very undesirable for our scheme in terms of use of the infrastructure nodes, and just change the values in the equations as explained.

Under those conditions, we will find out that the achievable throughput is

$$T=(1-\epsilon_n)\alpha\, k\,\frac{\log((n+m)q)}{\log(n+m)}\times\log\left(1+\frac{a_n\cdot\beta_n}{\frac{\sigma^2}{P}+(k-1)\,\mu_\gamma}\right) \tag{13}$$

by taking the expression in (7) and changing the network's size and the probability of good link. Proceeding the same way with the conditions included in the main result in [3], we will see that most of them will be relaxed. The most important change is our new upper bound to the number of simultaneous transmissions:

$$k=\alpha(n+m)\frac{\log((n+m)q)}{\log(n+m)} \tag{14}$$

We will also show that this scheme has an improved throughput, particularly whenever $\frac{\sigma^2}{P}-\mu_\gamma>0$, since in this case the throughput is maximized by choosing the biggest *k* and this scheme can bear a higher *k*. However, we will also show this approach to be suboptimal and thus we do not analyze further cases, but introduce a second and third scheduling approaches.

a) Vertex-disjoint paths construction and usage

First of all, it is interesting to take a closer look to the vertex-disjoint paths in [4] and the way they are used in [3]. Given a graph *G(n,p),* such as the one that forms an ad hoc network with random connections, the algorithm in [4] finds for each pair of nodes (a_i, b_i) a path connecting a_i to b_i, such that the set of paths so found is vertex-disjoint. Authors of [4] assume that the graph is chosen first, then the set of pairs to be united and last the paths.

It is important to note that the algorithm is subject to some restrictions:

1. all pairs must be disjoint (i. e. a node cannot be source and/or destination of many pairs at the same time).
2. Only a fraction of the nodes can be path endpoints[8], while the rest of them act as relay nodes or aren't used.
3. For a given path endpoint, only a fraction of its direct neighbors can be endpoints too.

Condition 2 is crucial for our further work. More concretely, the maximum number of pairs that are allowed in order to be able to obtain a set of vertex-disjoint paths is

$$k \leqslant \alpha n \frac{\log(np)}{\log n} \tag{4}$$

Now, imagine that the random graph *G(n,p)* contains *n/2* sources and *n/2* destinations that are paired. Using the algorithm in [4], one is just able to find vertex-disjoint paths for *k* sources and their *k* destinations at a time. In order to apply the algorithm to a network with *n/2* sources and *n/2* destinations, one needs to perform the following steps:

Step 1. Form *n/2* source-destination pairs (a_i, b_i) with *i* going from 1 to *n/2*.

Step 2. Chose a subset of the set of pairs defined in step 1, containing at most *k* pairs.

Step 3. Execute the algorithm in [4] for the given graph (network) and the subset of *k* pairs formed in step 2. As a result, one will obtain a set of vertex-disjoint paths binding those pairs. None of those *k* paths will share any node. The maximum length within those paths is the diameter of the network: $\frac{\log(np)}{\log n}$.

8 A endpoint is the node that acts as source or destination in path.

Step 4. Transmit the messages from a_j to b_j for all pairs of the subset chosen in step 2 using the paths obtained in 3. Perform all k transmissions simultaneously[9]. The duration of this step corresponds to the length of the longest path found in 3, which is always smaller or equal than $\frac{\log(np)}{\log n}$.

Step 5. Now that the communication between k pairs of nodes has been completed, chose a new subset of pairs of the original set in 1, ruling out the pairs that have already completed the communication.

Step 6. Repeat Steps 3 to 5 until all pairs have completed the communication.

We will now obtain an expression of the throughput that one can achieve in a hybrid ad hoc network by using this scheduling, although it is manifest that the usage of the nodes is very low, which is the main drawback of this scheme. This first approach runs the steps as described above although we will show k and h to have different values.

b) Number of hops and simultaneous transmissions

Since we have now $(n+m)$ nodes, the diameter of the network and the average distance between nodes are growing as $\frac{\log(n+m)}{\log((n+m)q)}$. In this case, and since we are constructing vertex-disjoint paths in a graph with $(n+m)$ vertex, the maximum number of simultaneous transmissions we can perform is

$$k \leqslant \alpha(n+m)\frac{\log((n+m)q)}{\log(n+m)} \tag{15}$$

and the number of hops that each message performs is

$$h = \frac{\log(n+m)}{\alpha\log((n+m)q)} \tag{16}$$

Since our nodes cannot transmit and receive at the same time, k is also bounded by $n/2$.

Note that although the number of hops is reduced, the number of simultaneous transmissions is still limited by the use of the random algorithm of [4] to find the paths, and it

9 Note that one can perform all transmissions at the same time without collision risk because the paths are disjoint, i. e. none of the nodes is used by more than one path.

restricts the improvement of the term *k/h* in the throughput's expression (3). The use of such scheduling is mainly based on [3]. However, as we will see later, one will be able to allow even larger number of simultaneous transmissions by applying the algorithm in [4] just locally or even ruling it out, making an overall improvement possible.

In [3], authors claim that considering that the number of interfering nodes is *(k-1)* is a pessimistic assumption, since in one time slot *t* some of the messages might already have reached their destination or been lost at a previous hop, and thus the number of interfering nodes would be smaller. In our case, we also have to take into account that the infrastructure nodes don't cause interference if they are communicating with another infrastructure node,

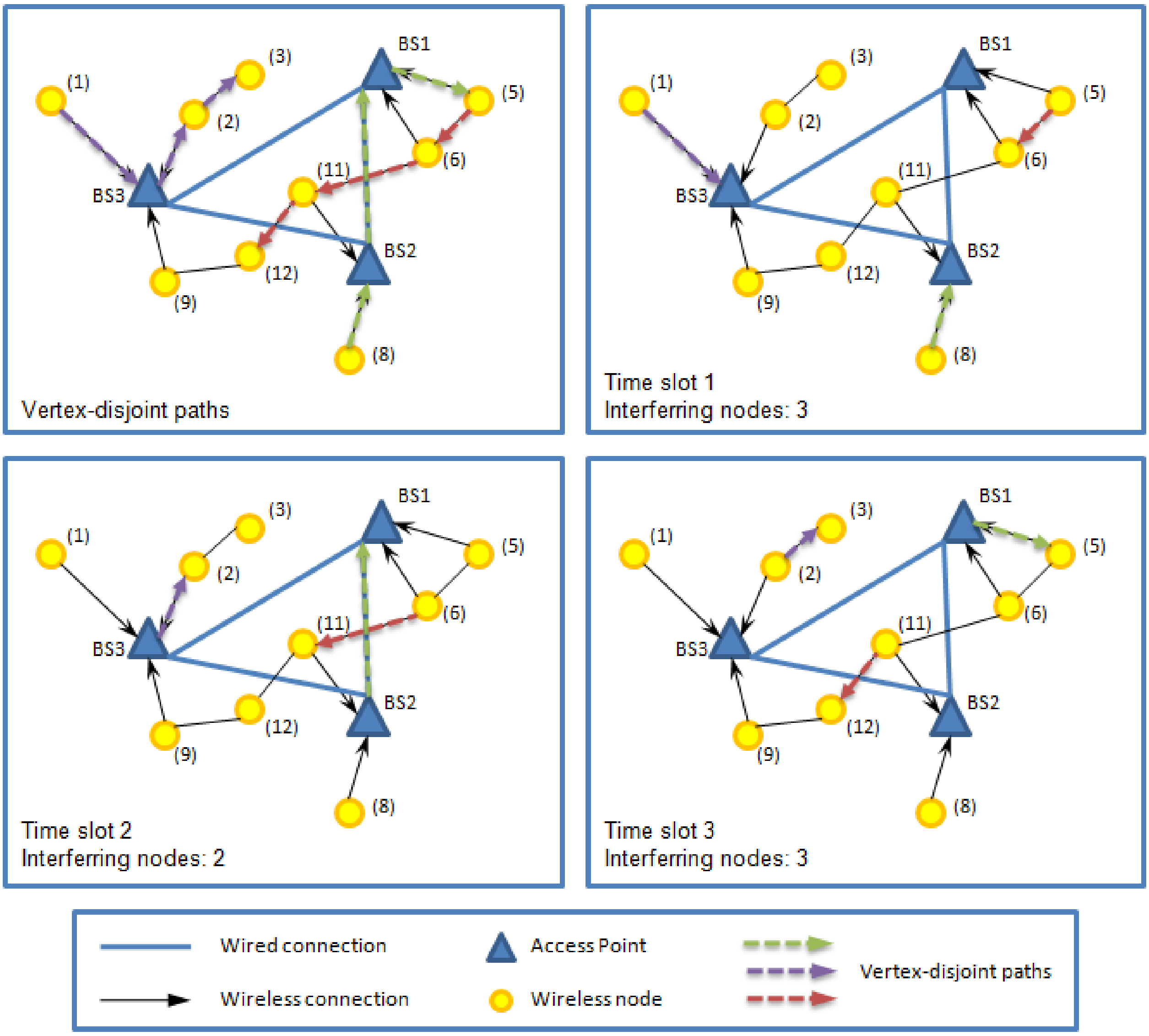

Fig. 6: Vertex disjoint paths and simultaneous transmissions during step 5

because the links between them are wired, and consequently the interference is even smaller in our network model.

The upper left scheme in Figure 6 illustrates the vertex-disjoint paths obtained in step 4, which bind a subset of 3 pairs source-destination. By the execution of the algorithm, 3 vertex-disjoint paths with the same length, 3 hops, have been obtained. The fact that they all have the same length is a coincidence though it always has to be smaller or equal than 4, which is the network diameter in this case. Note that none of the nodes are crossed by more than one path.

After that, nodes proceed to the transmission during 3 time slots. If the three paths wouldn't have the same length, the number of time slots needed would be the length of the longest path. The remaining 3 illustrations of Figure 6 show the performance of those transmissions in step 5. One can see that the green path follows a cellular path, while the other two paths follow an ad hoc path[10]. Thus, the green path causes less interference. During the first and the third time slot, all transmissions are wireless, thus the number of interfering nodes is 3. But the green path jumps from BS2 to BS1 in step 2, which are connected by a wired link, and thus it doesn't cause interference. In conclusion, the number of interfering transmissions in step 2 is 2, rather than 3 (which is the number of actual simultaneous transmissions). Also note that this drawing is optimistic in the sense that almost all nodes are used in the construction of the paths (only node (9) is excluded).

c) Probability of error

The probability of having a successful transmission between source and destination is the probability that all the hops in the chosen path meet the needed SINR. If E_t is the event of a link meeting the required SINR, we can express that as follows:

$$P(successful)=P\left(\bigcap_{t=1}^{h} E_t\right) \tag{17}$$

In our new network model, we have two kind of events: E_{ut} is the event of the wireless (unwired) link in hop t meeting the needed SINR and E_{wt} is the same for a wired one, which occurs with probability 1. We will note E_{ut}^{c} the complementary event of E_{ut} .

10 The purple path is an ad hoc path rather than a hybrid path because, although it uses an infrastructure node as relay node, it doesn't have any hop over a wired link.

Consequently, in a path with *h* hops, we have two kind of hops: wired (h_w) and unwired (h_u) obviously meeting $h = h_w + h_u$.

Taking these facts into account, we have that:

$$P(successful) = P\left(\bigcap_{t=1}^{h} E_t\right) = P\left(\bigcap_{t=1}^{h_u} E_{ut}\right) = 1 - P\left(\bigcup_{t=1}^{h_u} E_{ut}^c\right) \geqslant 1 - h_u P(E_{u1}^c) \tag{18}$$

Since the first hop is always wireless for every transmitting node, the number of interfering nodes in this hop is always larger or equal than in the rest of time slots and thus $P(E_{u1}^c) \geqslant P(E_{ut}^c)$ for any *t>1*. If we consider a pure ad hoc network with the same number of simultaneous transmissions the probability $P(E_{u1}^c)$ of the first hop not meeting the needed SINR in our hybrid network is equal to the probability of any hop not meeting the SINR in the pure ad hoc network[11]. Therefore, and since $h > h_u$ the probability of having a successful transmission is larger in the hybrid network than the one achieved in the pure ad hoc network. All told, the condition on ε is relaxed.

d) Throughput

Considering that, in the worst case, all other transmitting nodes *(k-1)* are causing interference, our new throughput becomes:

$$T = (1 - \epsilon_n) \alpha \, k \frac{\log((n+m)q)}{\log(n+m)} \times \log\left(1 + \frac{a_n \cdot \beta_n}{\frac{\sigma^2}{P} + (k-1)\mu_\gamma}\right) \tag{13}$$

which is obtained applying (2) and (16) to (3).

Recall from (7) that, in a pure ad hoc network, the throughput scales as:

$$T = (1 - \epsilon_n) \alpha \, k \frac{\log(n Q_n(\beta_n))}{\log n} \times \log\left(1 + \frac{a_n \cdot \beta_n}{\frac{\sigma^2}{P} + (k-1)\mu_\gamma}\right) \tag{7}$$

11 In a pure ad hoc network, all simultaneous transmissions cause interference in every hop, because all hops are wireless. This means that, considering that the length of the paths is constant, the amount of interference is the same in all hops and it is also the same in the first hop of the wireless network, when all simultaneous transmissions cause interference too.

Thus, when one schedules the same amount of simultaneous transmissions (k) and both networks have the same conditions (signal power, threshold, connections strength distribution, etc.), the gain achieved by the addition of the infrastructure is:

$$G=\frac{\log((n+m)q)\log n}{\log(n+m)\log(np)} \tag{19}$$

which is always bigger than 1. Note that having the same conditions for both networks, the gain only depends on the network's size, the number of infrastructure nodes and the connectivity.

To see the influence of p and m on the gain, we study Figure 7, which shows the evolution of the gain G as a function of the number of infrastructure nodes when $n=1000$ and $p=0.1$, $p=0.5$ and $p=0.9$ respectively. The number of ad hoc nodes is chosen to be relatively small (considering that we are interested in $n\rightarrow\infty$) as a pessimistic value, and the scale of m is chosen to have a maximum of $n/10$ infrastructure nodes[12].

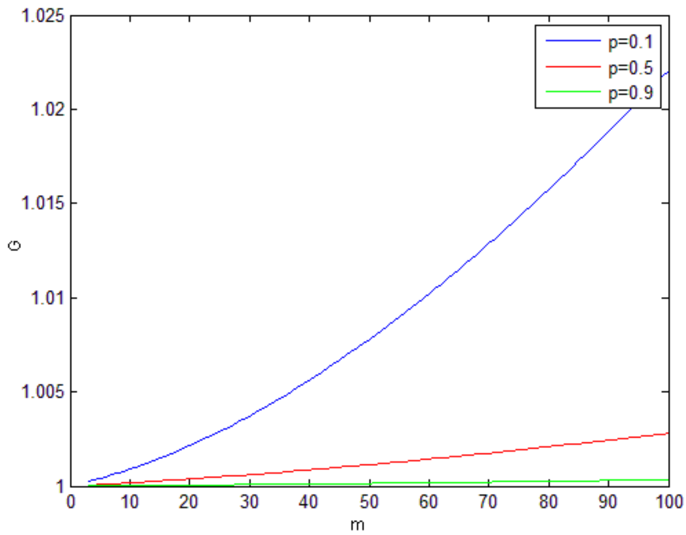

Fig. 7: Evolution of the gain as a function of m when n=1000

As we can see, the gain is bigger and increases more rapidly when the connectivity is smaller. Notice that this doesn't mean that the throughput of the hybrid network is inverse proportional to the connectivity, but that the improvement introduced by the addition of the

12 The size of the infrastructure should always be relatively small in comparison to the size of the network to be a realizable investment. In fact, it would be desirable to have m increasing sub-linearly with n.

infrastructure is more notable when the connectivity is low. Anyway, one can see that the improvement is small, which is the reason that we will introduce a second and third approach to the analysis.

e) *Gain when* $\frac{\sigma^2}{P}-\mu_\gamma>0$

Before we step to the next approach, we want to point out that the gain is higher whenever $\frac{\sigma^2}{P}-\mu_\gamma>0$ due to the fact that the hybrid network can bear a higher number of simultaneous transmissions in this case.

Appendix I shows that, when $\frac{\sigma^2}{P}-\mu_\gamma>0$, the expression of the throughput is non-decreasing with *k*, which means that selecting the biggest *k* is optimal. In that case, in the pure ad hoc network, the throughput is optimized by choosing $k=\alpha n\frac{\log np}{\log n}$, which is the biggest value possible (recall that the limit is fixed in (4)).

In our network, however, the maximum number of simultaneous transmissions that one can allow is fixed in (15) to be $k=\alpha(n+m)\frac{\log((n+m)q)}{\log(n+m)}=\alpha n\frac{\log((n+m)q)}{\log(n+m)}+\alpha m\frac{\log((n+m)q)}{\log(n+m)}$. Thus, our achievable throughput will be higher if this maximum value of *k* is higher than the one obtained for the pure ad hoc network.

Our new expression for *k* is equal to the old one when *m=0*. Thus, if we prove that this expression is non-decreasing with *m>0*, we will have proved that *k* can be made higher in the hybrid network than in the pure ad hoc one.

In a word, we want to prove that

$$\frac{\partial k}{\partial m}=\frac{\partial}{\partial m}\left[\frac{\log((n+m)q)}{\log(n+m)}\right]+\frac{\log((n+m)q)}{\log(n+m)}+m\frac{\partial}{\partial m}\left[\frac{\log((n+m)q)}{\log(n+m)}\right]>0 \quad ,$$

which is the same than

$$(m+1)\frac{\partial}{\partial m}\left[\frac{\log((n+m)q)}{\log(n+m)}\right] > \frac{-\log((n+m)q)}{\log(n+m)}$$

where $0 < \frac{\log((n+m)q)}{\log(n+m)} < 1$ [13] and *m>0*.

We need to assure that $\frac{\partial}{\partial m}\left[\frac{\log((n+m)q)}{\log(n+m)}\right]$ is positive or has a smaller absolute value than $\frac{\log((n+m)q)}{(m+1)\log(n+m)}$. Recall from (10) that *q* depends on *m*.

$$\frac{\partial}{\partial m}\left[\frac{\log((n+m)q)}{\log(n+m)}\right] = \frac{\frac{\log(n+m)}{(n+m)q}(nq' + q + mq') - \frac{\log((n+m)q)}{n+m}}{\log^2(n+m)} = .$$

$$. = \frac{q\log(n+m) - q\log((n+m)q) + (n+m)q'\log(n+m)}{(n+m)q\log^2(n+m)} = \frac{q\log\left(\frac{1}{q}\right) + (n+m)q'\log(n+m)}{(n+m)q\log^2(n+m)}$$

The denominator is positive, since every factor in it is positive too. The first term of the numerator is positive too, since *q* is positive, and the logarithm of a number bigger than 1 is always positive too.

To prove that the derivative above is positive, we still need to prove that the second term of the numerator is positive. First, note that *(n+m)>>1* and thus *log(n+m)>0*. On the other hand, the derivative of *q* can be found using (10):

$$q = p + \frac{m\cdot(m-1)}{((n+m)(n+m-1))}(1-p) = p + \frac{(m^2-m)}{(n^2+m^2+2mn-n-m)}(1-p) = .$$

$$q' = (1-p)\frac{[(2m-1)\cdot(n^2+m^2+2mn-n-m)-(m^2-m)\cdot(2m+2n-1)]}{[(n+m)^2(n+m-1)^2]} = ...$$

$$. = (1-p)\frac{[(2m-1)\cdot(n^2+2mn-n)+(m^2-m)]}{[(n+m)^2(n+m-1)^2]}$$

This last expression is also positive when *n* and *m* are at least 1, which always happens.

---

13 Recall that $q > p \geqslant \frac{\log n}{n}$ and thus $(n+m)q > nq > np > \log n > 1$ which means that $\log((n+m)q) > 0$.

We have proved that the expression is non-decreasing with *m* and thus, based on the results of Appendix I, our throughput will always be higher in a hybrid network when $\frac{\sigma^2}{P} - \mu_\gamma > 0$.

f) Considerations on this approach

There is still some work left undone in this approach. The number of wired and wireless hops that a message performs remains to be characterized, and so does the average number of interfering nodes. The probability of error also depends on the number of wired and wireless hops, although we have proved that the results regarding the probability of error in [3] build an upper bound for the probability of error in the hybrid network.

In conclusion, we have proven that adding infrastructure nodes always improves the throughput, though the gain in (19) is relatively small, and relaxes the restrictions on *ε*, *p* and others. We have also proven that whenever $\frac{\sigma^2}{P} - \mu_\gamma > 0$ the gain is higher than the one shown in (19).

However, since the vertex-disjoint paths algorithm of [4] used by [3] (and subsequently by our approach in this section) is highly sub-optimal, we continue with next scheme of scheduling rather than working further on this one.

Section V. Second scheduling approach for hybrid network: forcing the paths to use the infrastructure and enforcing vertex-disjoint paths

As we already stated, using the algorithm of [4] to find the paths is highly sub-optimal in terms of nodes usage. The algorithm also keeps the system from crossing any infrastructure node in more than one path, leading to a waste of infrastructure resources. Also note that, since it chooses the hops randomly and equiprobable from the graph formed by the nodes and the good links between them, one isn't giving priority to using wired links. However, it would be desirable to give priority to the usage of them due to their higher capacity and in order to reduce the interference. Also, the random choice of the paths can lead to cases where a message performs several hops over the infrastructure network although in the wired network any infrastructure node is capable to reach any other one by performing just one hop. Indeed, it makes sense to claim that forcing some paths to cross the infrastructure will reduce the number of hops per message compared to choosing them randomly, not only in the case explained above.

In this section,

(1) we show that, by using a second scheduling approach that restricts the number of hops to be less than $2 \cdot h_{max}$ and locally enforces the vertex-disjoint paths concept, we can achieve a throughput of:

$$T = (1-\epsilon)\frac{m}{2\tilde{h}}\log\left(1+\frac{a_n\beta_n}{\frac{\sigma^2}{P}+(m-1)\mu_\gamma}\right) \tag{20}$$

where $\tilde{h}$ is the average number of hops performed per message.

(2) We also show that the minimum number of infrastructure nodes needed to limit the number of hops as above is approximately

$$m \geqslant \frac{np-1}{p(np)^{h_{max}}} \tag{21}$$

(3) We detail the steps to be performed to construct the paths that lead to the achievable throughput in (20).

a) Maximum number of hops and access points' density

Suppose that we add an infrastructure to our network, which contains a fixed number of access point that are strongly connected, meaning: from any access point we can reach any other one using just one time slot. Also assume there are enough infrastructure nodes to guarantee that almost every ad hoc node is capable of reaching any infrastructure node in h_{max} hops or less. For this, the nodes are to be divided within what we will call sub-graphs around every access point, where every node is at h_{max} hops from the access point or less. If we consider a hops metric instead of the geometrical layout of the network, we can imagine the infrastructure node as the center of set of concentric circumferences with radius 1, 2, 3, ..., h_{max} hops containing the nodes that can reach to it by performing 1, 2, 3, ..., h_{max} hops respectively. The infrastructure node is also the center of an imaginary circle of radius h_{max} hops that contains every node within h_{max} hops of the access point, i. e. every node in the sub-graph. In this case the diameter of every sub-graph is $2 \cdot h_{max}$. Note that the circumference's notion isn't related to the geometrical placement of the nodes.

Since the number of hops is necessarily an integer, the nodes of the subgraph are placed in concentric circumferences with radius 1 to h_{max}. We will name C_i^j the circle centered at infrastructure node *j* with radius *i,* which contains all nodes within *i* hops from infrastructure node *j*; c_i^j the circumference (which we will also name ring) centered at the *j*-th infrastructure node with radius *i*, which contains the nodes exactly *i* hops away from the infrastructure node *j*; and N_i^j and n_i^j are the number of nodes within the circle and its circumference respectively. We will also say a node is in the *i-th* level if the shortest path from that node to the closest access point takes exactly *i* hops. The number of nodes in level *i* is

$$n_i = \sum_{j=1}^{m} n_i^j \tag{22}$$

and

$$N_i = \sum_{j=1}^{m} N_i^j \tag{23}$$

is the number of nodes within *i* hops from any access point. Note that the model necessarily meets $N_{h_{max}} = n$ and that $N_{h_{max}}^j = n_j'$ is the number of nodes in a sub-graph *j*.

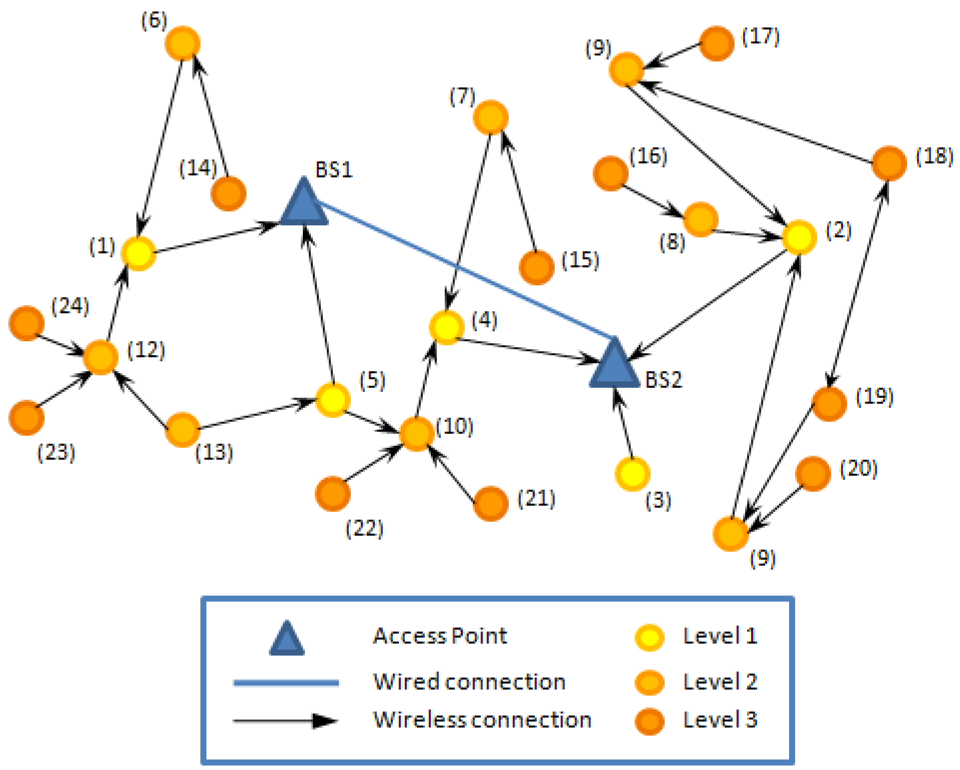

Fig. 8: Levels and circumferences around infrastructure nodes

Figure 8 shows how levels and circumferences are formed depending on the number of hops within the node and the nearest access point and the independence of them with the geographical distribution of the nodes. Let's take node (5) as an example: note how it is in the first level because it has a direct connection to infrastructure node BS1, but it can also reach BS2 through node (10). Thus, node (5) would be in level 3 if BS1 or the connection to it wouldn't exist. Also note how it is connected to BS1 but not to node (4) nor node (22), though nodes (4) and (22) are geometrically nearer than BS1. Also note how, for instance, node (10) is placed in level 2 no matter which infrastructure node it tries to reach, because it is 2 hops away from either of them, thus node (10) belongs both to c_2^1 and c_2^2 . This same reason causes nodes (21) and (22) to be contained both in c_3^1 and c_3^2 .

Tables 1, 2 and 3 summarize the number of nodes in every circle, circumference and level to clarify those definitions. Note that the sum of nodes in the circumferences might be larger than the total number of nodes in the level, because a node, like node (10) for instance, can belong to more than one circumference and circle. This shows how circles and circumferences are not disjointed.

Circumference radius (i)	Infrastructure node (j)	
	j=BS1	j=BS2
i=1	2	3
i=2	4	5
i=3	5	8

Table 1: Number of nodes in every circumference

Circle radius (i)	Infrastructure node (j)	
	j=BS1	j=BS2
i=1	2	3
i=2	6	8
i=3	11	16

Table 2: Number of nodes in every circle

Level	Number of nodes
1	5
2	8
3	16

Table 3: Number of nodes in every level

From random graphs theory[14], we know that the number of adjacent vertices to a certain node (nodes directly connected through a good link to a certain node) is close to *np*, the number of vertices/nodes at distance 2 is also close to $(np)^2$, and the number of nodes at distance k is approximately $(np)^k$. Thus, the number of vertices within distance k from a

14 See chapter VII in [6] for reference.

infrastructure node *j* is approximately $N_k^j \approx \sum_{i=1}^{k} (np)^i$. Recall from (12) that $p > \frac{\log n}{n}$ and thus *np>log n>1*, which means that we can sum

$$N_k^j = \frac{(np)^{k+1} - np}{np - 1} \tag{24}$$

When *k*=h_{max}, N_k^j becomes the number of nodes contained the *j*-th sub-graph (n_j'). Since, as stated before, the number of adjacent nodes to any node in the network is expected to be tightly close to *np,* we can also expect the number of nodes in every sub-graph to be tight, i. e. $n'_{j1} \approx n'_{j2}$ for any $j_1 \neq j_2$, we will assume that all sub-graphs contain the same number of nodes: $n'_{j1} \approx n'_{j2} \approx n'$, and thus:

$$n' \approx \frac{(np)^{h_{max}+1} - np}{np - 1} \tag{25}$$

Since *np* is expected to be big and h_{max} is a positive integer also expected to be bigger than 1, we can approximate:

$$n' = \frac{np(np^{h_{max}} - 1)}{np - 1} \approx \frac{(np)^{h_{max}+1}}{np - 1} \tag{26}$$

hence

$$h_{max} \approx \frac{\log(n'(np-1))}{\log np} - 1 \tag{27}$$

Since *m* is the number of sub-graphs, every sub-graph contains *n'* nodes and the sub-graphs are not necessarily disjoint, we must have $n < m \cdot n'$, which means $m \geqslant \frac{n}{n'}$ and using (25) for *n'* we get the following relation:

$$m \geqslant \frac{n(np-1)}{(np)^{h_{max}+1}-np} = \frac{np-1}{p((np)^{h_{max}}-1)} \tag{28}$$

which proves expression (21) if we approximate $(np)^{h_{max}}-1 \approx (np)^{h_{max}}$. Also, as we expected, for a certain fixed graph (having given *n* and *p*), the number of necessary access points increases as we decrease h_{max}.

b) Scheduling with local use of vertex-disjoint paths

The second scheduling approach is based on the idea of forming vertex-disjoint paths locally within every sub-graph rather than overall the network. Those paths are to be found using the same steps introduced in Section IV, executing them at the same time in every sub-graph. The transmission is going to be performed in three phases:

Phase 1. Vertex-disjoint paths are found within the sources sub-graphs connecting the source to the infrastructure node. Note that if the target of the communication is within the same source's sub-graph, one could directly find an ad hoc path within the sub-graph, which could also use the infrastructure node as relay node, but not the wired links that connects them to other infrastructure nodes. However, to simplify the algorithm and in order to bound the duration of this phase to h_{max}, we will just assume that every message crosses some infrastructure node.

After finding a path, the source will transmit its message to phase's 1 end point, i. e. the closest infrastructure node. This phase is performed using the steps described in Section IV at the same time in every sub-graph, and repeated as many times as needed, until all sources have sent their messages to the access point.

Phase 2. In phase 2, the infrastructure nodes exchange all the messages that need to perform hybrid paths to go from source to destination, i. e. nodes that need to change to another sub-graph. Since infrastructure nodes are connected by high capacity links and are capable of handling many transmissions at the same time, one assumes this phase to be performed in one time slot.

Phase 3. In phase 3, which is performed symmetric to phase 1, the infrastructure nodes transmit the messages to the final destinations, again using vertex-disjoint paths. Note that one can re-use the paths found in phase 1 to perform this one, just following them in the opposite direction.

This separation in phases assures that, by avoiding collisions inside every sub-graph, we avoid collisions in the whole network.

With this scheme, we improve the number of hops and reduce the interference by forcing sources that aren't in the same sub-graph as their destination to use a path that crosses the infrastructure network. If the destination node is within the same source's sub-graph, it will be reachable in at most $2 \cdot h_{max}$ hops that will be performed between phases 1 and 3. Otherwise, a hybrid path is drawn, going in phase 1 from the source to a infrastructure node related to the source, then reaching the destination's related infrastructure nodes in one hop in phase 2 and from there to the destination within h_{max} more hops in phase 3. Thus we can be sure that any source can reach any destination in $2 \cdot h_{max}+1$ hops.

Figure 9 illustrates the explanation above with two paths. The upper one belongs to the communication between two nodes that are placed in different sub-graphs and as far from their closest access points as possible. In the drawn network, the maximum number of hops within any node and the closest infrastructure node is 3. In the upper illustration, one can see that, to communicate with node (20), node (24) needs to perform 7 hops during the three phases: 3 hops to reach BS1, then one hop to reach BS2 from BS1 and last 3 hops to reach the destination from BS2. The illustration below shows the performance of the phases when both source and destination belong to the same sub-graph. In this case, phase 2 is eliminated. Note that indeed a shorter ad hoc path between nodes (24) and (22) could be found but, as mentioned above and to simplify the scheduling scheme, we won't consider this option.

Moreover, the transmissions between infrastructure nodes performed in phase 2 cannot collide with the wireless transmissions of phases 1 and 3 because they are wired. Therefore, phase 2 can begin while phase 1 is in process and phase 3 can also begin while the second one is in course if the first one has been completed (to keep avoiding collisions inside every sub-graph). Thus, we can consider that phase 2 doesn't consume time resources.

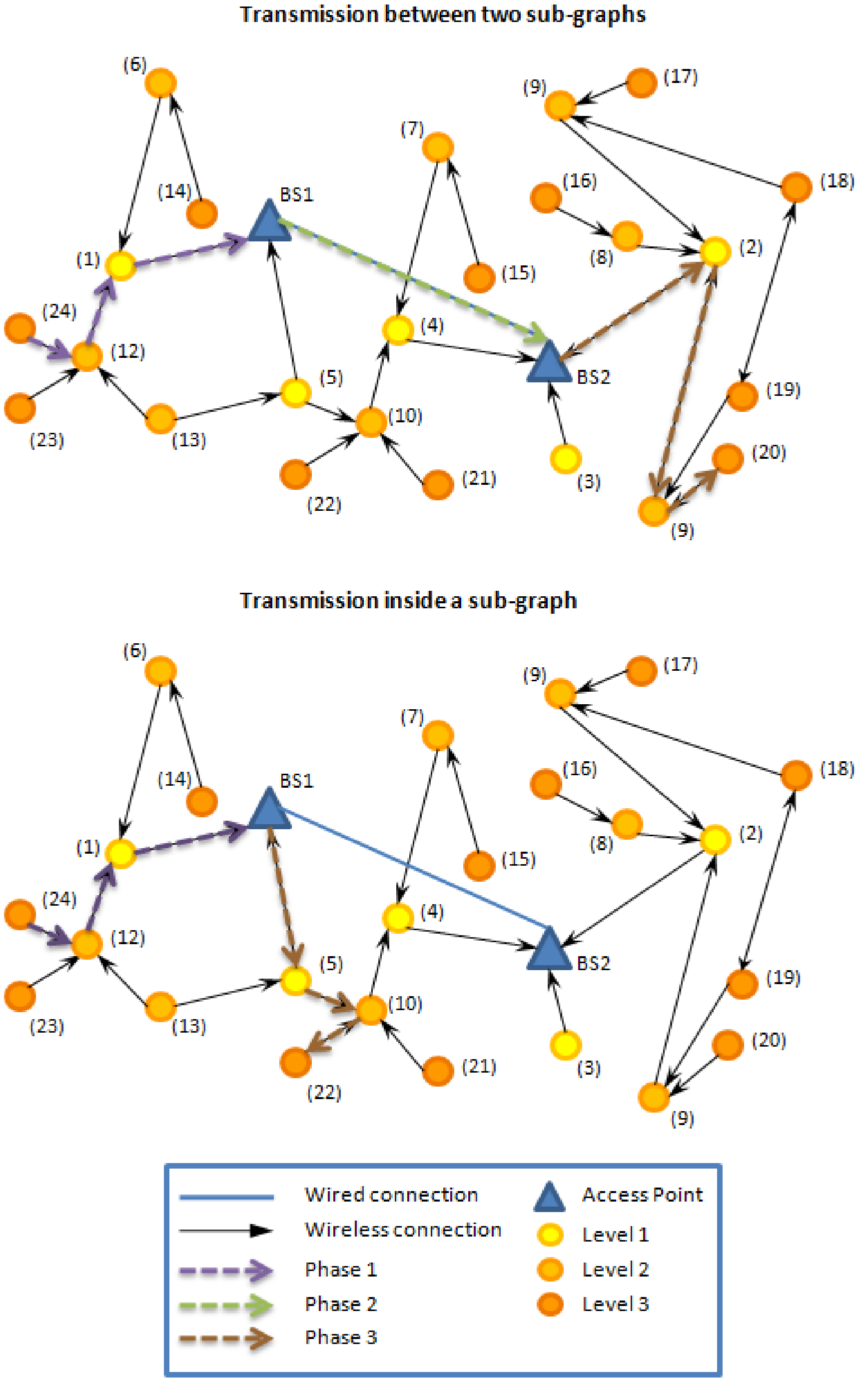

Fig. 9: Transmission phases

Note that this scheduling improves the number of hops if and only if $2 \cdot h_{max}+1$ is smaller than the network's diameter, i. e. $2\,h_{max}+1<\frac{\log(n+m)}{\log((n+m)q)}$. However, since the transmissions in the second phase are wired, the phase causes no interference, and so the SINR is always improved with regard to what is explained in Section IV for any *k*, which is decisive in interference limited networks.

c) Worst case throughput

The proposed scheme causes a bottleneck in every infrastructure node, because these nodes need to be used in every transmission[15], but infrastructure nodes can only receive one wireless transmission at one time. In the worst case all sources are h_{max} hops away from any infrastructure node and choose a destination outside of their sub-graph that is also h_{max} hops away from any access point, thus the maximum time needed for the transmission is $2 \cdot h_{max}$, and every transmission needs to use the infrastructure. Moreover, if we want to use vertex-disjoint paths, we can only have at most *m/2* simultaneous transmissions, having one source or destination in every sub-graph.

But indeed, since an infrastructure node is capable of managing many transmissions at the same time in its wired connections, we can relax the restrictions for the number of simultaneous transmissions found before and allow every sub-graph to contain one source and one destination, as phases 1 and 3 aren't allowed to overlap in time. In this case, we can schedule *m* simultaneous transmissions. Putting these values and using the definition (2) of the SINR threshold in expression (3), the achievable throughput is:

$$T=(1-\epsilon)\frac{m}{2\,h_{max}}\log\left(1+\frac{a_n\beta_n}{\frac{\sigma^2}{P}+(m-1)\mu_\gamma}\right) \tag{29}$$

Isolating h_{max} from (28) with *np>>1* we have approximately

15 The effect of also routing through the infrastructure nodes the messages that could perform an ad hoc path inside the sub-graph is considered negligible, because the number of nodes inside a sub-graph is expected to be small in relation to the network's size, and thus the number of messages that suffer this case is expected to be proportionally small.

$$h_{max} \geqslant \frac{\log(np-1) - \log(mp)}{\log np} = \frac{\log\left(\frac{np-1}{mp}\right)}{\log np} \tag{28 bis}$$

Then by substituting in (29) we have the following bound:

$$T = (1-\epsilon)\frac{m \cdot \log np}{2\log\left(\frac{np-1}{mp}\right)} \log\left(1 + \frac{a_n \beta_n}{\frac{\sigma^2}{P} + (m-1)\mu_\gamma}\right) \tag{30}$$

Note that the usage of the wired nodes is at most three time slots (once in every phase) of the total transmission time i. e. $\left(\frac{3}{2\,h_{max}}\right)$, even if all the nodes need to use them. This value can be low if h_{max} is relatively high. Therefore, it would be desirable to find a way to schedule transmissions such that we can exploit the infrastructure resources better.

d) Achievable throughput considering the average number of hops

The previous study finds an expression of the achievable throughput. However, it is interesting to point out that the presented network scenario, where all sources are as further from the infrastructure nodes as possible, with maximum number of hops needed, is an extreme and improbable case. Hence, next we use in the analysis the average number of hops between any node and the closest infrastructure node, $\tilde{h}$, and find the corresponding throughput.

As we have already introduced, the number of nodes in c_i^j is $n_i^j \approx (np)^i$. Since the rings and sub-graphs are not necessarily disjoint from other rings or sub-graphs, we find an upper bound to the number of nodes in every level as follows: in the first level, we will find $n_1 \leqslant m \cdot np$ nodes and every one of those has approximately *np* neighbors, thus $n_2 \leqslant m(np)^2$ and $n_i \leqslant m(np)^i$. Note that n_i is always increasing with *i*.

The average number of hops that a message needs to perform is:

$$\tilde{h}=\sum_{i=1}^{h_{max}} i\frac{n_i}{n}\leqslant\frac{1}{n}\sum_{i=1}^{h_{max}} i\cdot m\cdot(np)^i \tag{31}$$

If p is not a function of n, we can further calculate $\tilde{h}$ as:

$$\tilde{h}\leqslant\frac{1}{n}\sum_{i=1}^{h_{max}} i\cdot m\cdot(np)^i=\frac{mnp}{n}\sum_{i=1}^{h_{max}} i\cdot(np)^{i-1}=mp\sum_{i=1}^{h_{max}}\frac{\partial}{\partial n}(np)^i$$

$$\frac{\partial}{\partial n}\sum_{i=1}^{h_{max}}(np)^i=\frac{\partial}{\partial n}\frac{(np)^{h_{max}+1}-np}{np-1}=\frac{(np)^{h_{max}}(h_{max}+1)(np-1)-p((np)^{h_{max}+1}-1)}{(np-1)^2}$$

Again, since $np>\log n$ is big and $h_{max}>1$, we can approximate $(np)^{h_{max}+1}-1\approx(np)^{h_{max}+1}$ and:

$$\frac{\partial}{\partial n}\sum_{i=1}^{h_{max}}(np)^i\approx\frac{(np)^{h_{max}}(h_{max}+1)(np-1)-p(np)^{h_{max}+1}}{(np-1)^2}=(np)^{h_{max}}\frac{(h_{max}+1)(np-1)-np^2}{(np-1)^2}$$

$$\tilde{h}\leqslant mp(np)^{h_{max}}\frac{(h_{max}+1)(np-1)-np^2}{(np-1)^2} \tag{32}$$

During phases 1 and 3 we find a path from the node to the infrastructure node using a relay node in every circumference c_i. Since in the worst case all paths end in the infrastructure node (i. e. all sources chose a destination outside their sub-graph), we avoid collisions by only having one transmission at a time inside every sub-graph, leading to a bound of m simultaneous transmissions (one per sub-graph). Since we can only have one transmission in every sub-graph at one time, the higher the number of sources in a sub-graph is, the longest the time needed to complete the first phase in that particular sub-graph. The first phase's duration equals the longest time needed a sub-graph amongst all sub-graphs to complete it. The maximum number of sources inside any sub-graph is n', thus the first phase's duration is maximum whenever we have at least one sub-graph with all n' nodes acting as sources. This maximum is given by:

$$t_{max,1}=n'\tilde{h} \tag{33}$$

In phase 3, the longest time also corresponds to the case where a sub-graph contains only destinations, and the time needed to perform it is, following the same reasoning, $t_{max,3}=t_{max,1}$.

This means that, for the worst case, we need $2 \cdot t_{max,1}$ time slots to send, performing the three phases, messages from n sources to n destinations[16]. Note that the maximum number of simultaneously transmitting nodes that we can have in every sub-phase is m, although the more different is the number of source and destinations in a sub-graph, the less number of simultaneous transmissions we can perform and the smaller the interference will be. However, for analysis simplicity, we consider $(m-1)$ interfering nodes. Hence, using $k=n$ transmissions, $h=2 \cdot t_{max,1}$ time slots are needed to complete the three phases, and the definition of (2) for the SINR threshold expression, in (3), the achievable throughput is:

$$T=(1-\epsilon)\frac{n}{2\,t_{max,1}}\log\left(1+\frac{a_n\beta_n}{\frac{\sigma^2}{P}+(m-1)\mu_\gamma}\right)=(1-\epsilon)\frac{n}{2n'\tilde{h}}\log\left(1+\frac{a_n\beta_n}{\frac{\sigma^2}{P}+(m-1)\mu_\gamma}\right)$$

Since $m\geqslant\frac{n}{n'}$ the achievable throughput is bounded by:

$$T=(1-\epsilon)\frac{m}{2\tilde{h}}\log\left(1+\frac{a_n\beta_n}{\frac{\sigma^2}{P}+(m-1)\mu_\gamma}\right)$$

which proves the claim in (20).

e) Considerations on this approach

In this section, we have introduced a mechanism to force the scheduling to use the infrastructure with the aim of reducing the interference and the number of hops that a message needs to perform to get to its intended destination. Note that this scheme also works if one doesn't use the vertex-disjoint algorithm inside the sub-graphs, because having

16 This scheme allows us for every node to be source and destination of a message, due to the separation in phases.

just one transmission inside a sub-graph during every phase is enough to warrant that collisions won't happen.

The main limitation of this scheme still resides in the usage of vertex-disjoint paths, which is inherited from the approach presented in [3]. Next section analyzes the performance of an approach that rules out this idea, and allows many simultaneous transmissions in every sub-graph.

Section VI. Third scheduling approach for hybrid network: forcing the paths to use the infrastructure and ruling out the vertex-disjoint paths

The throughput in the previous sections is limited by the number of simultaneous transmissions that one is allowed to perform in a certain time slot. This restriction is forced by the condition that one allows a single transmission in every sub-graph to assure that messages don't collide by the application of vertex-disjoint paths, though in fact, this condition is much stronger than our needs.

In the following we assume a scheduling that rules out the idea of using vertex-disjoint paths and allows many simultaneous transmissions in every sub-graph and time slot, for which we specify the corresponding limitations and the resulting throughput:

$$T=(1-\epsilon)\frac{m}{2}\log\left(1+\frac{a_n\beta_n}{\frac{\sigma^2}{P}+(m\tilde{h}-1)\mu_y}\right) \tag{34}$$

a) Scheduling

At the beginning of Section V, we have seen that nodes and infrastructure can be conceived as sub-graphs and circles. Note that this conceiving implies that a message originated in a node placed in ring c_i^j needs to perform *i* hops choosing a relay node in every circle c_{i-1}^j to c_1^j in order to reach the closest infrastructure node. As we already stated, forcing the paths to cross the infrastructure gives an accurate limit of the paths length that depends on the nodes' level.

In this section, we will assume that one knows a scheduling procedure where paths are found choosing in every hop a node in the internal ring. We will also assume that the scheduling doesn't cause collision, this is a node isn't chosen as destination of a certain hop if it has already been chosen by another one for the same time slot.

b) Number of simultaneous transmissions and throughput

Let's focus on a certain hop, for instance the hop between c_i^j to and c_{i-1}^j . This has to be performed by all the messages originated in c_i^j , but also by the messages originated further from the infrastructure node. For instance: in a network with 5 levels, all messages originated in levels 3, 4 and 5 will eventually have to perform the hop from level 3 to level 2, but only messages originated in levels 4 and 5 need to perform the hop from level 4 to 3. Thus, the closer to the infrastructure node a hop is, the higher the number of messages that need to perform it. Moreover, the last hop (from c_1^j to the access point) needs to be performed by every message.

On the other hand, since the population of the circles decreases when we approach the infrastructure node, the number of non-occupied nodes in the destination ring (potential receivers) is always smaller than the number of nodes in the source ring. This means that the number of simultaneous messages that can perform a certain hop is always limited by the number of non-occupied nodes in the internal ring, if we take into account that a node cannot receive two messages at the same time or transmit and receive a message at the same time. In every time slot, the maximum number of nodes in level 1 that can transmit to all infrastructure nodes is m, and the first level still has n_1-m non-occupied nodes that can receive n_1-m messages when is n_1 the total number of nodes in level one. That is, the maximum number of messages that can jump from level 2 to 1 is n_1-m, and following the same reasoning, n_2-(n_1-m) messages can jump from level 3 to level 2.

Origin	Destination	# of simultaneous hops	# of messages that perform the hop
c_1	BS	$n_0=m$	$n \approx m \sum_{i=1}^{h_{max}} (np)^i$
c_2	c_1	$n_1-m \approx mnp-m$	$n-n_1 \approx m \sum_{i=2}^{h_{max}} (np)^i$
c_3	c_2	$n_2-(n_1-m) \approx mnp^2-mnp+m$	$n-n_1-n_2 \approx m \sum_{i=3}^{h_{max}} (np)^i$
c_{i+1}	c_i	$(-1)^i m + m \sum_{j=1}^{i} (-1)^{i-j} (np)^j$	$m \sum_{j=i+1}^{h_{max}} (np)^j = m \frac{(np)^{h_{max}+1} - (np)^{i+1}}{np-1}$

Table 4: Number of simultaneous hops allowed in every level and number of messages that need to perform each of them

We summarize the behavior of a sub-graph in Table 4, where in the third column we use the fact that $n_i \approx m(np)^i$.

Figure 10 illustrates this behavior:

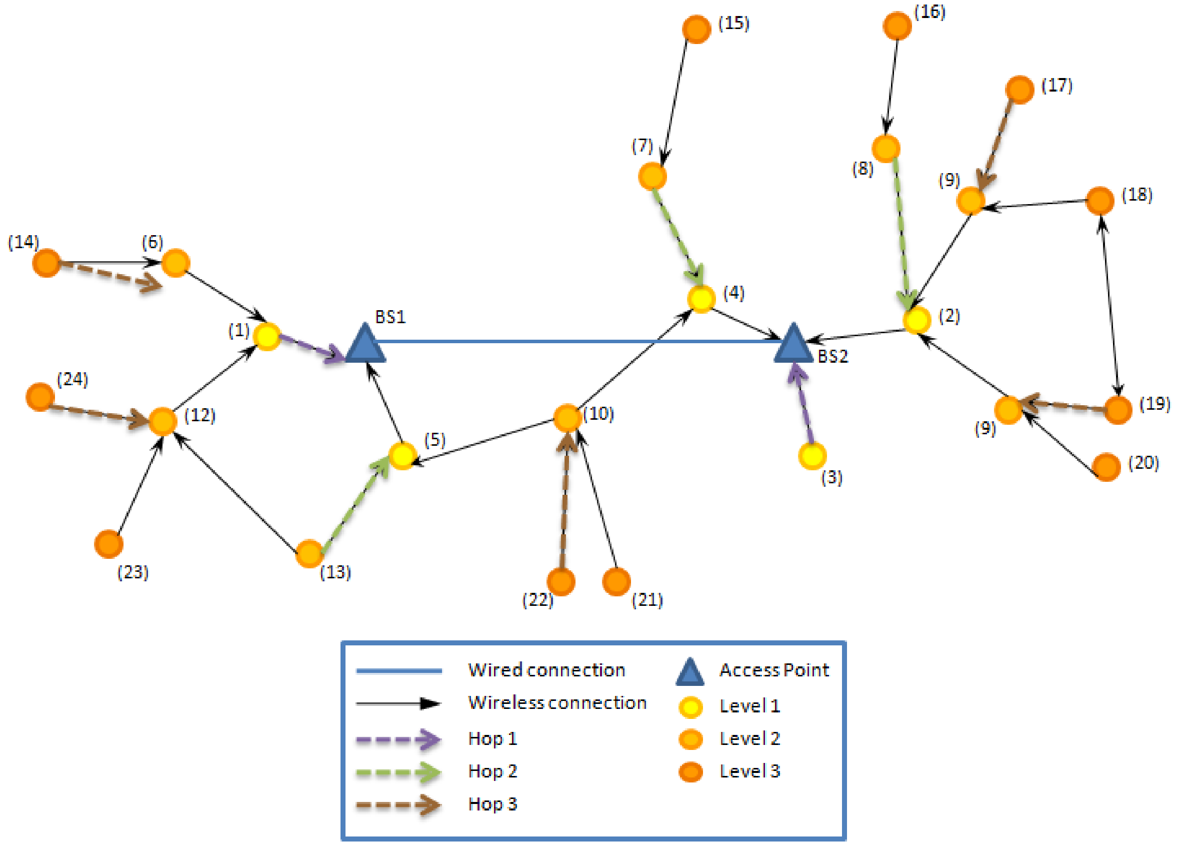

Fig. 10: Network topology in a hops metric and simultaneous hops

As we see in Figure 10, the number of transmissions that can be scheduled at a certain level is limited by the number of nodes in the lower level. For instance, since there are 5 nodes in level 1 but only 2 infrastructure nodes, only nodes (1) and (3) can simultaneously transmit towards the infrastructure. In the same way, since level 1 has 5 nodes and 2 of them are busy transmitting to the infrastructure, one can schedule 3 simultaneous transmissions from level 2 to 1.

Note that from the table above one can show that the number of simultaneous hops in any circle is always bigger than *m* if *n>100* (see appendix II for proof), which is not a restriction since we are studying $n \to \infty$. Furthermore, the number of messages that have to perform a certain hop is bigger for levels closer to the infrastructure. Again in Figure 10, one can see that n_1=5, n_2=8 and n_3=11. Messages originated in level 3 (11 messages) need to perform hops from level 3 to level 2, then from level 2 to level 1 and finally from level 1 to the infrastructure. Analogously, messages originated in level 2 (8 messages) jump to level 1 and then to the infrastructure node and messages from level 1 (5 messages) jump directly to the infrastructure. As a summary, there are just 11 messages that need to perform a hop from level 3 to level 2 but all 24 messages need to perform a hop from level 1 to the infrastructure. As stated before, the congestion increases when rings are closer to the access point.

This means that the capability of the network to bring messages from level *i+1* to level *i* is always bigger than the capability to perform the next hop from level *i* to *i-1* and, since there is only one infrastructure node in every sub-graph, there is a bottleneck in the last hop, from level 1 to the infrastructure. Therefore, the total time for *n* messages to complete the first phase is *n'* (the time for all the messages in a sub-graph to perform the last hop) and the total time to complete the three phases is *2n'*, and so the achievable throughput is:

$$T=(1-\epsilon)\frac{n}{2n'}\log\left(1+\frac{a_n\beta_n}{\frac{\sigma^2}{P}+(k-1)\mu_\gamma}\right) \tag{35}$$

where *k* is the number of simultaneous transmissions. In general, we can say that we have to perform $n\tilde{h}$ hops in *n'* time slots, thus the average number of simultaneous hops performed in every time slot has to be: $k\approx\frac{n\tilde{h}}{n'}\leqslant m\tilde{h}$. Recall that the number of simultaneous transmissions in every time slot is a parameter fixed by the scheduling used, thus we can adapt our schedule to maintain a balanced number of simultaneous transmissions in every time slot, and optimize the minimum SINR needed. Also note that this bound of *k* isn't a system requirement, but a bound to our needs, i. e. the number of simultaneous transmissions that the network needs to handle to optimize this scheme is at most $m\tilde{h}$. In conclusion, the throughput is bounded by:

$$T=(1-\epsilon)\frac{m}{2}\log\left(1+\frac{a_n\beta_n}{\frac{\sigma^2}{P}+(k-1)\mu_\gamma}\right) \tag{36}$$

which is a proof of claim in (34).

In Appendix III, we prove that if $\frac{\sigma^2}{P} - \mu_\gamma \geqslant 0$ the optimum choice is the biggest *m* possible. Note that it is necessary to put other restrictions on the number of simultaneous transmissions to delimit the probability of error, and we will choose the number of infrastructure nodes based on those parameters. For economical reasons, it makes no sense to add a number of access points bigger than the number of simultaneous transmissions that our system can bear.

c) Probability of error

Now we analyze the probability of error for this last result. Following the proof in [3], let's point out that the number of hops that a message performs is $h=h_u+h_w \leq 2 \cdot h_{max}+1$ where the message performs at most one wired hop and the number of wireless hops that a message has to perform is smaller or equal to $2 \cdot h_{max}$. Like in previous sections, we consider that the wired connections are ideal, which have no transmission errors. Thus, using the same notation as in (18), the probability that a message fails to be sent is bounded as:

$$P(successful) = P\left(\bigcap_{t=1}^{h} E_t\right) = P\left(\bigcap_{t=1}^{h_u} E_{ut}\right) = 1 - P\left(\bigcup_{t=1}^{h_u} E_{ut}^c\right) \geqslant 1 - P\left(\bigcup_{t=1}^{2h_{max}} E_{ut}^c\right) \geqslant 1 - 2h_{max} P(E_{u1}^c) \tag{37}$$

Again, in [3], we find a proof that

$$P(E_t^c) \leqslant \frac{\sigma^2 / (k-1)}{\left(\frac{P\beta_n - \rho_0 \sigma^2}{(k-1) P \rho_0} - \mu_\gamma\right)^2} \tag{38}$$

which in our case (and considering the case where we have a higher interference in which $k \approx m\tilde{h}$) leads to:

$$P(successful) \geq 1 - 2h_{max} P(E_t^c) \geq 1 - 2h_{max} \frac{\sigma_\gamma^2 / (m\tilde{h}-1)}{\left(\frac{P\beta_n - \rho_0 \sigma^2}{(m\tilde{h}-1)P\rho_0} - \mu_\gamma\right)^2} \quad (39)$$

We will need this probability to tend to one, meaning that the second term should go to zero.

d) Comparison of the different achievable throughputs

In Sections V to VI we have introduced a set of different possible throughputs, based on the use of different scheduling. Next, we will compare their expressions and requirements. Note that those different expressions only differ in the number of simultaneous transmissions that every scheme allows us to handle and the time needed to complete the transmission.

We summarize those results in Table 5:

Case	Achievable throughput	Simultaneous transmissions
1. Worst case (one transmission per sub-graph and maximum *h*)	$T=(1-\epsilon)\frac{m \cdot \log np}{2\log\left(\frac{np-1}{mp}\right)}\log\left(1+\frac{a_n\beta_n}{\frac{\sigma^2}{P}+(m-1)\mu_\gamma}\right)$ (30)	$k = m$
2. One transmission per sub-graph and average number of hops	$T=(1-\epsilon)\frac{m}{2\tilde{h}}\log\left(1+\frac{a_n\beta_n}{\frac{\sigma^2}{P}+(m-1)\mu_\gamma}\right)$ (20)	$k = m$
3. Simultaneous transmissions in every sub-graph	$T=(1-\epsilon)\frac{m}{2}\log\left(1+\frac{a_n\beta_n}{\frac{\sigma^2}{P}+(k-1)\mu_\gamma}\right)$ (34)	$k \leq m\tilde{h}$

Table 5: Achievable throughputs by the algorithms in Sections V and VI

First of all, note that if *m* is large enough, leading to a layout where every wireless node can directly communicate with the access point, all the expressions above are equivalent:

$$T=(1-\epsilon)\frac{m}{2}\log\left(1+\frac{a_n\beta_n}{\frac{\sigma^2}{P}+(m-1)\mu_\gamma}\right) \quad (40)$$

by scheduling *m* simultaneous transmissions. However, note that the amount of interference in this case is going to be high, due to the fact that *m* needs to be high to achieve the condition of having just one level. The expression (40) is monotonous increasing with *m* if $\frac{\sigma^2}{P}-\mu_\gamma>0$. Thus, whenever this condition is accomplished, we can increase the throughput by adding more infrastructure nodes.

Next, recall that the expression (30) represents a worst case scenario that is improbable. The improvement introduced by (20) with regard to (30) is caused by a refinement of the analysis but not to a improvement of the network behavior.

Last, if we allow various simultaneous transmissions in a sub-graph at the same time, we get the achievable throughput in (34). In this case, the interference in the network is higher, which means that allowing multiple transmissions in a sub-graph might not always be optimal. In fact, the gain of case 3 over case 2 is:

$$G=\tilde{h}\frac{\log\left(1+\frac{a_n\beta_n}{\frac{\sigma^2}{P}+(k-1)\mu_\gamma}\right)}{\log\left(1+\frac{a_n\beta_n}{\frac{\sigma^2}{P}+(m-1)\mu_\gamma}\right)} \tag{41}$$

and it is bigger than 1 if and only if:

$$\log\left(1+\frac{a_n\beta_n}{\frac{\sigma^2}{P}+(m-1)\mu_\gamma}\right)^{\tilde{h}}<\log\left(1+\frac{a_n\beta_n}{\frac{\sigma^2}{P}+(k-1)\mu_\gamma}\right) \tag{42}$$

In conclusion, the number of infrastructure nodes and the connectivity will determine which of both schedulings is optimal. In a interference limited network, chances are that it will be preferable to have only one transmission in every sub-graph and meet a higher SINR.

Comparing the performance of these two models and the one introduced in Section IV is complex due to the high number of factors that have influence in the three approaches and remains undone in our work. However, it is easy to see that approaches in Sections V and VI introduce improvement by reducing the amount of interference when having the same number of transmissions and also allowing a higher number of simultaneous transmissions.

Section VII. A tighter model to the network topology

The calculations in Sections V and VI do not take into account the fact that a node might be contained in several circles (in a different level and/or around another infrastructure node). This means that the number of nodes at the first level is smaller than *mnp* since some nodes might be in more than one circle of the first level.

Furthermore, in the next level there are two sources of error by considering $m(np)^2$: the overlap between the circles but also the number of nodes in the first level, whose immediate neighbors are to be added to the second level. As we can see, the error increases in every level, leading to a sub-estimation of the number of needed infrastructure nodes and an over-estimation of the number of nodes in every circle and sub-graph. Figure 8[17] shows this phenomenon. One can see how node (5) is connected to the infrastructure node BS1 and to node (10), and it belongs to level 1 rather than to level 3. In the same way, node (18) is connected to nodes (9) and (19), which belong to levels 2 and 3 respectively and thus node (18) belongs to level 3.

We will now introduce a mechanism to construct disjointed sub-graphs and calculate the expected number of nodes in a sub-graph, accordingly.

a) Circles' construction

To build disjoint circles, we will construct them as follows: starting at any access point, we build a ring c_1^1 containing all the nodes adjacent to it. Next, we move to another infrastructure node and build the first circle around it c_1^2 taking all the immediate neighbors (nodes directly connected to it) that aren't already contained in c_1^1 . We continue performing this operation for every access point *j*, including in c_1^j all nodes directly connected to j and not included in the previous circles c_1^1 ... c_1^{j-1} .

As we stated, although every infrastructure node is expected to be directly connected to *np* nodes, we have to take into account that a node can be directly connected to more than one

17 See page 41.

access point, thus the number of nodes directly connected to the infrastructure is smaller than *mnp*. The mechanism explained above constructs the first level assuring that no pair of circles share any node. To compute the expected number of nodes belonging to level 1 n_1 (i. e. the sum of the number of nodes in the circles constructed above) we will use the probability distribution of the number of nodes at level 1. We define p_1 as the probability that a certain node is contained in the first level. Then, the probability distribution of the number of nodes in the first level is binomial $B(n,p_1)$ (see the properties of a binomial distribution in Appendix IV):

$$P(k\,nodes \in n_1) = \binom{n}{k} p_1^k (1-p_1)^{n-k} \tag{43}$$

Note that p_1 is the probability that a certain node is in the first ring, which is also the probability that this node is directly connected to some access point, and the complementary of the probability that a node isn't connected to the infrastructure (this is, isn't connected to any access point at all). The event of a node not being connected to a certain access point[18] is independent from the non-existence of a *good connection* between this same node and all other access points, therefore the probability of non-existence of the m connections between a certain node and the m infrastructure nodes is $(1-p)^m$, and we can express p_1 as:

$$p_1 = p(\text{connected to some access point}) = 1 - p(\text{not connected to any access point}) = 1 - (1-p)^m \tag{44}$$

Due to the binomial distribution's properties, we can compute the expected value of n_1 as:

$$E[n_1] = n \cdot p_1 = n\left(1-(1-p)^m\right) = n - n(1-p)^m \tag{45}$$

and the variance[19] as:

$$\sigma_{n_1} = n \cdot p_1 (1-p_1) = n \cdot (1-p)^m \cdot (1-(1-p)^m) = n(1-p)^m - n(1-p)^{2m} \tag{46}$$

18 Recall that the probability that a certain connection exists is p and thus the probability of non-existence is $1-p$.

19 Note that the variance is always positive.

According to Chebychev's inequality, $P(|n_1 - E[n_1]| > 4\sigma_{n_1}) < \frac{1}{16} < 0.07$, thus, we can say that n_1 is, with high probability bounded by:

$$E[n_1] - 4\sigma_{n_1} < n_1 < E[n_1] + 4\sigma_{n_1} \qquad (47)$$

We are interested in the lower bound, that can be developed as:

$$E[n_1] - 4\sigma_{n_1} = n - 5n(1-p)^m + 4n(1-p)^{2m} \qquad (48)$$

which means that the distance between n and n_1, i. e. the number of nodes not contained in level 1, is with high probability high bounded by:

$$n - n_1 < 5n(1-p)^m - 4n(1-p)^{2m} \qquad (49)$$

Thus, we can conclude that there is a minimum connectivity p, which can assure that the first ring contains every node in the network, with high probability. As as proven in Appendix V, the number of nodes not contained in the first ring (n-n_1) is smaller than *1* if:

$$p < 1 - \sqrt[m]{\frac{5n - \sqrt{(5n)^2 - 16n}}{8n}} \qquad (50)$$

From (49), and as proven in Appendix VI, we can also conclude that, for a given network (i. e. for a given pair of n and p) there is a minimum number of infrastructure nodes for which the network has, with high probability, just one level:

$$m > \frac{\log\left(\frac{5n - \sqrt{(5n)^2 - 16n}}{8n}\right)}{\log(1-p)} \qquad (51)$$

In case that the first level doesn't contain all nodes in the network, one proceeds to form the second level in a similar way. To compute the expected number of nodes in the second level, one has to take into account that this second level is constructed given the first one (i. e.

given that the first level contains nodes n_1): the nodes in the second level are chosen from the remaining nodes $(n-n_1)$, and a certain node is in the second level if it is connected to some node contained in level c_1. The distribution of n_2 is also binomial, similar as before:

$$P(k\,nodes \in n_2/n_1) = \binom{n-n_1}{k} p_2^k (1-p_2)^{n-n_1-k} \tag{52}$$

where p_2 is the probability that a certain node is in the second level, given that the first level contains n_1 nodes. This probability is the complementary of not being directly connected to any node in c_1. Again, since the existence of *good connections* is independent, the probability that a certain node isn't connected to any of the n_1 nodes in the first level is $(1-p)^{n_1}$ and thus:

$$p_2 = 1 - p(\textit{not connected to any of the } n_1 \textit{ nodes in } c_1) = 1-(1-p)^{n_1} \tag{53}$$

$$E[n_2/n_1] = (n-n_1)(1-(1-p)^{n_1}) = n-n_1-(n-n_1)(1-p)^{n_1} \tag{54}$$

$$\sigma_{n_2/n_1} = (n-n_1)\cdot p_2(1-p_2) = (n-n_1)(1-p)^{n_1} - (n-n_1)(1-p)^{2n_1} \tag{55}$$

And the number of nodes not in the first two levels is with high probability bounded by:

$$n-(n_1+n_2) < 5(n-n_1)(1-p)^{n_1} - 4(n-n_1)(1-p)^{2n_1} \tag{56}$$

This means that for a given n_1 there is a minimum probability (smaller than the one found in (50)) for which the first two levels contain all the nodes with high probability. This threshold is:

$$p < 1 - \sqrt[n_1]{\frac{5(n-n_1) - \sqrt{(5(n-n_1))^2 - 16(n-n_1)}}{8(n-n_1)}} \tag{57}$$

One can define any arbitrary level $i > 1$ following the same steps:

$$P(k\,nodes \in n_i / n_1 .. n_{i-1}) = \binom{n - \sum_{j=1}^{i-1} n_j}{k} p_i^k (1-p_i)^{n - \sum_{j=1}^{i-1} n_j - k} \tag{58}$$

where p_i is the probability that a certain node is directly connected to some node from the last level *(i-1)* given that the previous levels contain $n_1 ... n_{i-1}$ respectively.

$$p_i = 1 - p(\text{not connected to any node in } c_{i-1}) = 1 - (1-p)^{n_{i-1}} \tag{59}$$

$$E[n_i / n_1 .. n_{i-1}] = \left(n - \sum_{j=1}^{i-1} n_j\right)\left(1-(1-p)^{n_{i-1}}\right) \tag{60}$$

$$\sigma_{n_i / n_1 .. n_{i-1}} = \left(n - \sum_{j=1}^{i-1} n_j\right) \cdot p_i (1-p_i) = \left(n - \sum_{j=1}^{i-1} n_j\right) \cdot \left(1-(1-p)^{n_{i-1}}\right)(1-p)^{n_{i-1}} \tag{61}$$

And thus the number of nodes not contained in the first *i* levels is bounded as:

$$n - \sum_{j=1}^{i} n_j < 5\left(n - \sum_{j=1}^{i-1} n_j\right)(1-p)^{n_{i-1}} - 4\left(n - \sum_{j=1}^{i-1} n_j\right)(1-p)^{2 n_{i-1}} \tag{62}$$

and the minimum probability for which the *i*-th level is the last one[20] is:

$$p < 1 - \sqrt[n_{i-1}]{\frac{5(n - \sum_{j=1}^{i-1} n_j) - \sqrt{\left(5(n - \sum_{j=1}^{i-1} n_j)\right)^2 - 16(n - \sum_{j=1}^{i-1} n_j)}}{8(n - \sum_{j=1}^{i-1} n_j)}} \tag{63}$$

b) Average number of hops with disjoint sub-graphs

We can use the lower bounds from (49), (56) and more generally (62) as a pessimistic approximation to the number of nodes in every level. Once we know the number of nodes

20 The *i*-th level is the last level if and only if the distance between *n* and the sum of nodes included in levels 1 to *i* is smaller than 1.

that every level contains, we can re-calculate the average number of hops between a node and the nearest access point $\tilde{h}$:

$$\tilde{h} = \frac{1}{n} \sum_{i=1}^{h_{max}} i \cdot n_i \tag{64}$$

The throughput formulas introduced in Sections IV, V and VI are still valid, and one may apply our new estimations for the number of nodes in every ring and the average number of hops to calculate a more accurate value of the achievable throughput.

Section VIII. Results

In this section, we will analyze the performance of our system for different scalings of *m* and probability distributions of β_n when using the approaches introduced in Sections V and VI. More concretely, we will consider *m* scaling as $m=\frac{n}{\log n}$ and $m=\sqrt{n}$. We will also analyze the minimum number of infrastructure nodes needed to have a network that has, with high probability, just one level, and for this case we will compare the network performance to the results of the previous work.

a) Network performance when $m=\frac{n}{\log n}$

First we will consider the number of access points growing as $m=\frac{n}{\log n}$, which corresponds to a sub-linear scaling in relation to *n*. Note that, due to economical reasons, it is desirable that *m* is considerably smaller than *n*. This scaling requires a relatively big amount of infrastructure nodes when *n* is small, but the ratio *m/n* decreases when *n* grows, thus it is interesting for our study when *n* goes to infinity.

Using the simple model introduced in Sections V and VI, the number of adjacent nodes to a certain node (nodes directly connected to it) is very close to *np* and thus the number of nodes in the first level would be $n_1 \approx mnp$. Recall from (12) that, in order to avoid isolated nodes, we required $p>\frac{\log n}{n}$ and thus for the chosen *m,* $mp=\frac{n}{\log n} p>\frac{n}{\log n}\frac{\log n}{n}=1$ or $mnp\approx n_1\approx n$. This means that, according to the simple model of Sections V and VI, by introducing $m=\frac{n}{\log n}$ access points we can say that every node is connected to some infrastructure node with high probability. However, in Section VII, we have seen how to define better the levels' behavior by equations (43)-(63). By the analysis of this tighter model, we next show that, for the given *m,* the probability that the first level contains every ad hoc node in the network is actually close to 1. The following plots illustrate this.

First, in Figure 11, one sees the behavior of the minimum probability needed to have just one level that contains every node in the network, as stated in (50). As one can see from the figure, this limit diminishes rapidly, specially for larger values of *n*, and it also has a low starting value. Thus, we can say that it is highly probable that every node is adjacent to some infrastructure node when $m=\frac{n}{\log n}$, in which case the network has just one level.

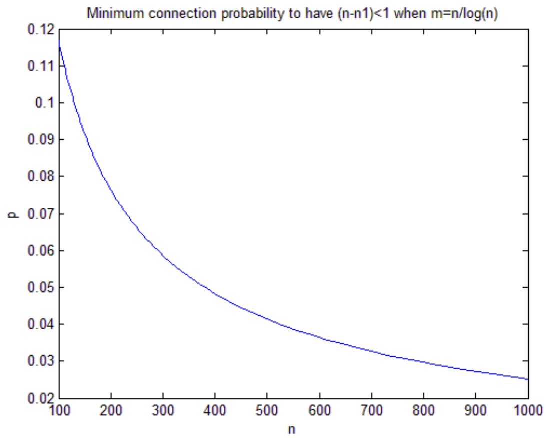

Fig. 11: Minimum probability to have $n-n_1<1$

Note from Figure 11 that, the larger our network is, the smaller connection probability we need to assure $h_{max}=1$, which means that a larger network has a better overall performance when adding this proportion of access points. Also remember, as stated in the analysis of Table 5, that whenever $h_{max}=1$ all schedulings proposed in Sections V and VI are equivalent.

Hence, whenever the network is not interference limited and the probability is big enough to guarantee that $h_{max}=\tilde{h}=1$, we conclude from (20)[21], by substituting $m=\frac{n}{\log n}$ and taking the highest number of simultaneous transmissions, that the throughput scales like:

$$T=(1-\epsilon)\frac{m}{2}\log\left(1+\frac{a_n\beta_n}{\frac{\sigma^2}{P}+(m-1)\mu_\gamma}\right)=(1-\epsilon)\frac{n}{2\log n}\log\left(1+\frac{a_n\beta_n}{\frac{\sigma^2}{P}+(\frac{n}{\log n}-1)\mu_\gamma}\right) \tag{65}$$

21 This result is also correct if we substitute the number of hops and access points in (30) or in (34).

The following plots illustrate the scaling of this throughput as a function of *n* with different values for the parameters $a_n \beta_n$, μ_γ and *SNR* (signal-to-noise ratio or σ^2/P) when the throughput is always increasing with *m.* Recall that β_n is the strength's threshold for a connection to be *good* and thus it is always smaller than 1 and that a_n is a factor smaller than 1. The plot values have been chosen taking into account that all of the factors are necessarily smaller than 1 and taking reasonably pessimistic suppositions[22].

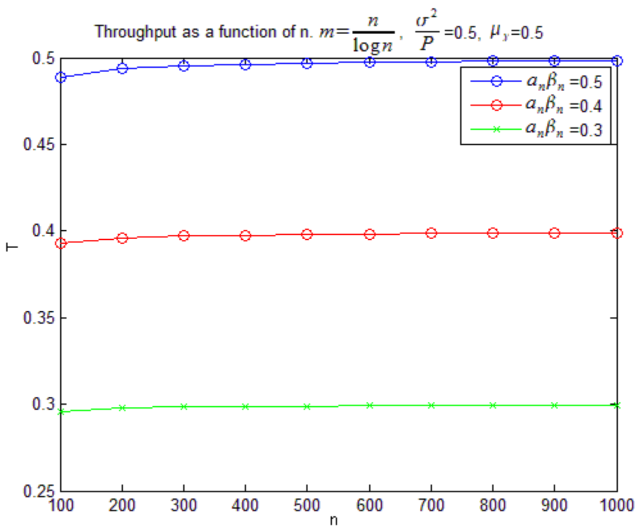

Fig. 12: Throughput for different values of $a_n \beta_n$

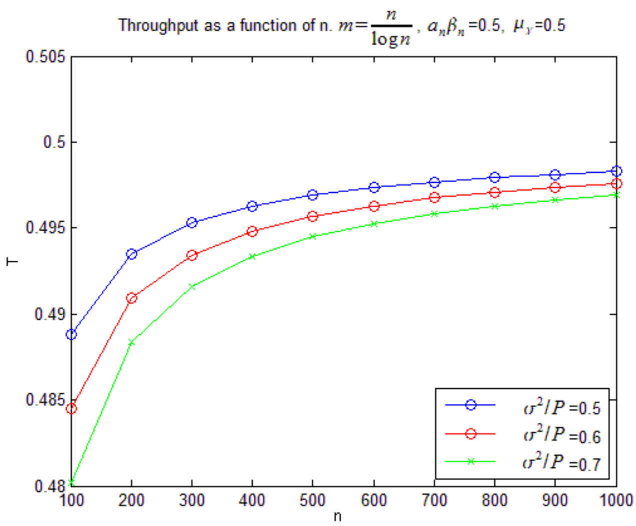

Fig. 13: Throughput for different values of σ^2/P

22 Note that having a large σ^2/P has a negative effect in the network, while $a_n \beta_n$ and μ_γ are desirable to be large.

As one can see from Figure 13, the *SNR* has only a slight impact on the throughput performance, which gets smaller as *n* increases, so that all curves tend to similar limits, enclosed between in a throughput variation range smaller than 0.005. On the other hand, from Figure 12, one notices that the product $a_n \beta_n$ can dramatically change the throughput performance, particularly at the limit with *n* where the curve tends. Surprisingly, Figure 14 indicates that the average strength of the connections has a great impact, wherein the throughput improves when it gets smaller.

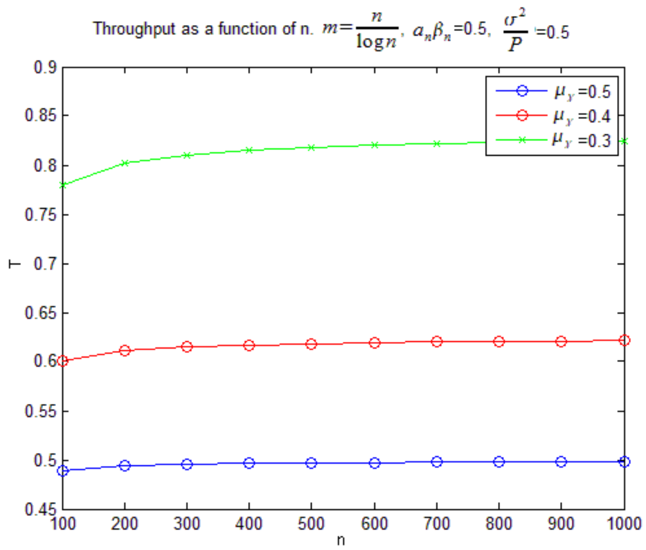

Fig. 14: Throughput for different values of μ_γ

Note, however, that the average strength μ_γ of the connections is related to the threshold β_n, thus there is a compromise between those two factors that will depend on the probability distribution of the connection strengths.

When the network's connectivity is not good enough, one will have more than one level, in which case the performance of the network depends on the chosen scheduling. Next, we analyze the achievable throughput in (34) when the average number of hops is 2 [23].

We repeat in Figures 15 and 16 Figures 12 and 14 with the same parameters except for having an average number of hops of 2. From corresponding comparison, it is easy to see the strong negative impact of increasing the number of levels (decrease of throughput) caused by an increase of the average number of hops.

23 Note that this means that number of levels is higher than 2.

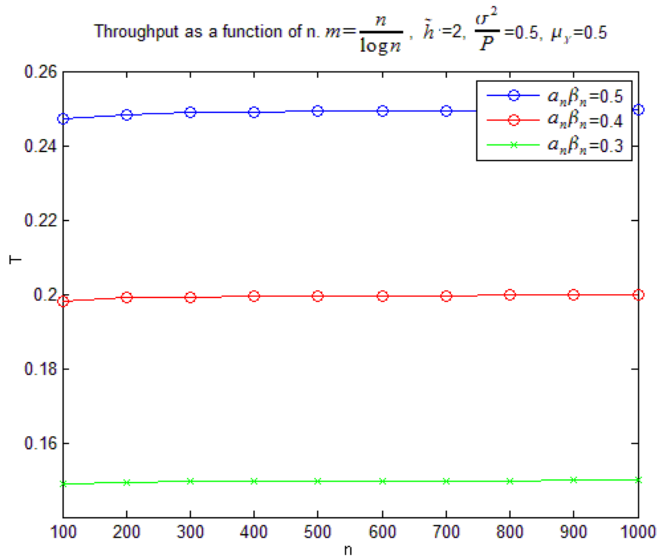

Fig. 15: Throughput for different values of $a_n\beta_n$ *and 2 levels*

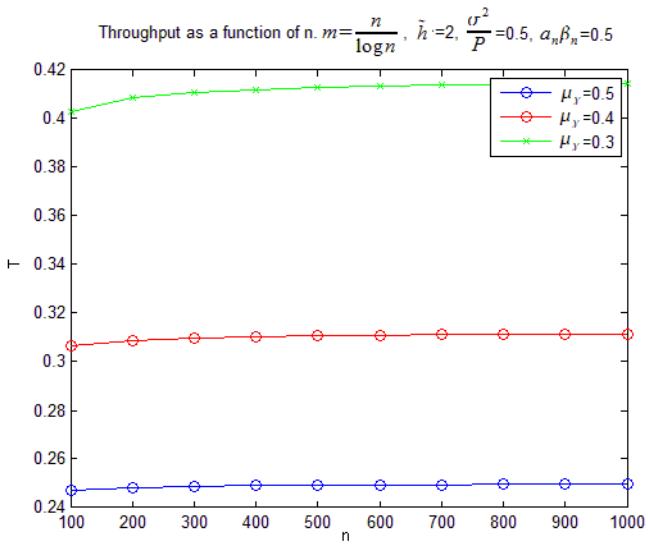

Fig. 16: Throughput for different values of μ_γ *and 2 levels*

b) Network performance when $m=\sqrt{n}$

We have seen that the performance of the network can be considerably improved by the addition of $m=\frac{n}{\log n}$ infrastructure nodes, if the network's connectivity is high enough.

However, we have also seen that this improvement worsens when the network has more than just one level. Next, we wonder if we can reduce the number of access points and still notice an improvement in the throughput of the network, thus we will repeat the previous analysis now considering the slower growth of $m=\sqrt{n}$.

Note from expressions (20) and (34) that the scaling of *m* has a linear impact in the throughput, besides of the effect of determining (together with *p*) the network topology and thus average number of hops. This means that by reducing *m* to $m=\sqrt{n}$ we are changing two factors with impact in the performance.

Figure 17 illustrates the curve of the minimum connectivity needed in order to have a network with just one level when *m* grows as the square root of *n*.

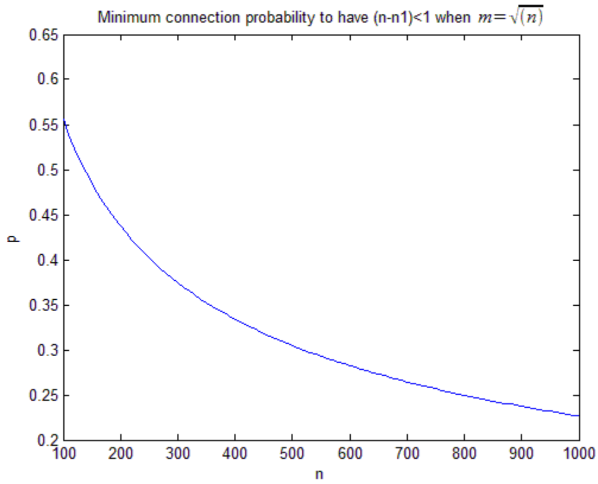

Fig. 17: Minimum probability to have n-n_1<1

By comparison to Figure 11, one easily notices that the requirements for having just one level in this case are much stronger, which is equivalent to state that the average number of hops is more likely to be higher than 1. However, it is interesting to point out that, although in Figure 17 the curve starts at a higher value than in Figure 11, this also decreases more rapidly, which means that, for some large enough *n*, the minimum connectivity needed to have just one level tends to be the same whether *m* scales as $m=\frac{n}{\log n}$ or $m=\sqrt{n}$.

In case that one has a network that accomplishes the required *p*, the achievable throughput according to (34) behaves like:

$$T = (1-\epsilon)\frac{m}{2}\log\left(1+\frac{a_n\beta_n}{\frac{\sigma^2}{P}+(m-1)\mu_\gamma}\right) = (1-\epsilon)\frac{\sqrt{(n)}}{2}\log\left(1+\frac{a_n\beta_n}{\frac{\sigma^2}{P}+(\sqrt{(n)}-1)\mu_\gamma}\right) \quad (66)$$

Repeating Figures 12, 13 and 14 for this throughput we get Figures 18, 19 and 20.

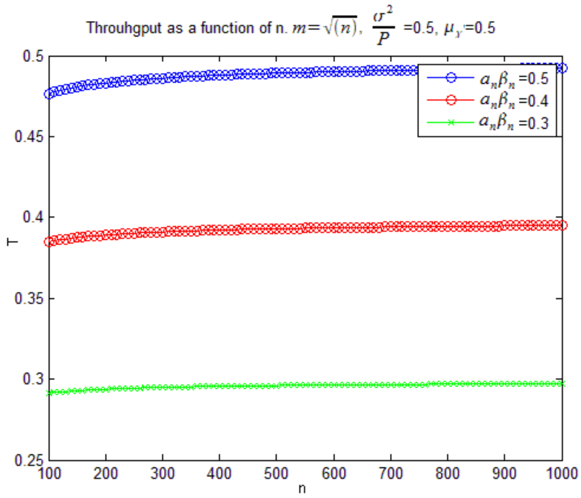

Fig. 18: Throughput for different values of $a_n\beta_n$

As one can see, the curves have similar values to Figures 12, 13 and 14, which means that changing $m=\frac{n}{\log n}$ to $m=\sqrt{n}$ almost doesn't change the achievable throughput whenever the average number of hops stays the same.

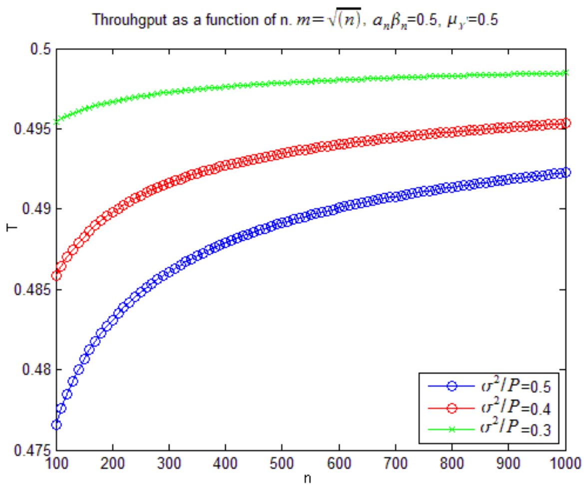

Fig. 19: Throughput for different values of σ^2/P

On the contrary, recall from Figures 11-16 that increasing the number of hops had a dramatical effect on the performance of the network. Thus, we can conclude that, the main influence of the scaling of *m* in the throughput resides in the determination of the number of levels and the population in them, rather than in the determination of the number of simultaneous transmissions that one can schedule.

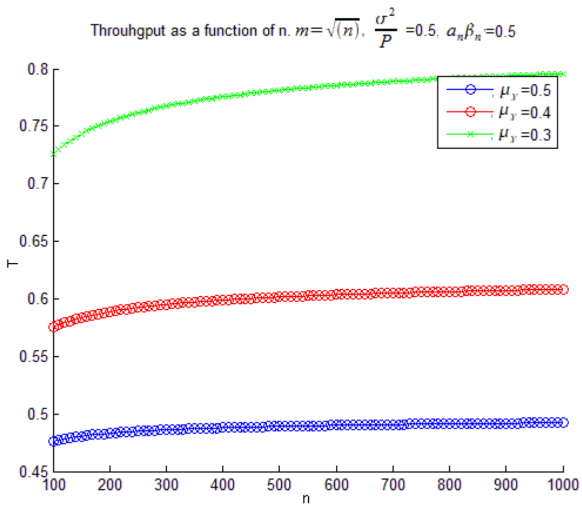

Fig. 20: Throughput for different values of μ_Y

c) *Minimum number of access points to have one level and overall performance compared to a pure ad hoc network*

In (51), we have found an expression for the minimum number of infrastructure nodes that one needs to have just one level. As seen above, the average number of hops has a critical influence on the performance, and thus this minimum number of access points is crucial.

Figure 21 illustrates the scaling of the minimum *m* for different small scalings of *p* (recall that, in the worst case, $p=\frac{\log n}{n}$).

One can see that for the worst value of *p*, the minimum number of access points is larger than the number of nodes in the network. However, by a linear scaling of the lower threshold of *p*, the number of infrastructure nodes needed is rapidly reduced. Note that, since *n* is expected to be large, this lower threshold is very small, and thus it isn't too optimistic to expect to have values such as $p=6\frac{\log n}{n}$ [24]. In fact, as one can appreciate in Figure 22, when taking p=0.1, which is still a considerably low value for the connectivity, the ratio between the number of nodes and the number of access points improves.

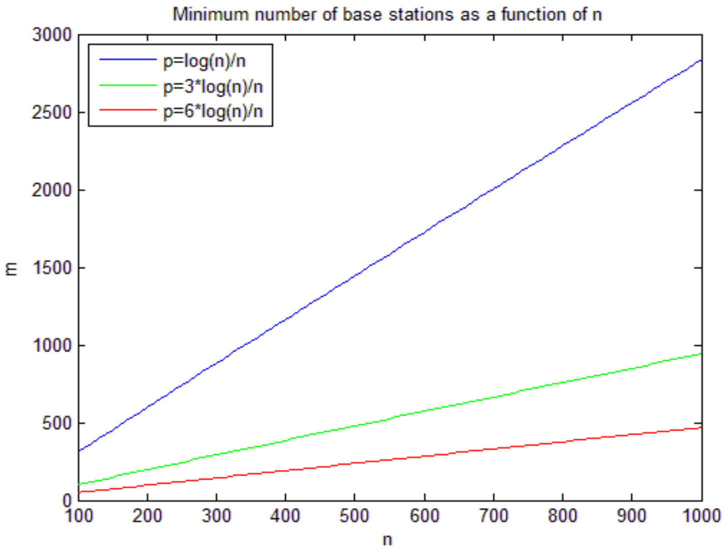

Fig. 21: Minimum number of access points s to have $\tilde{h}=1$ *for different values of p*

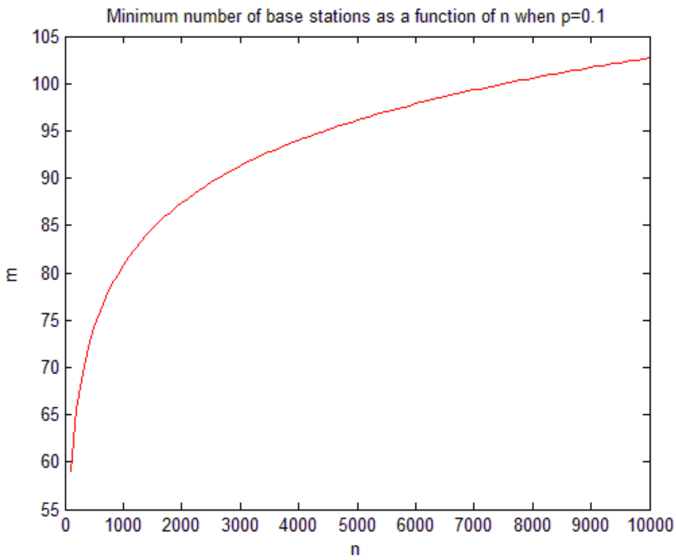

Fig. 22: Minimum number of access points to have $\tilde{h}=1$ *for p=0.1*

24 For instance, if *n=1000*, $\frac{\log n}{n}=0.003$ and $6\frac{\log n}{n}=0.018$

As stated above, whenever we have the minimum *m* needed to have just one level, the achievable throughput scales as:

$$T=(1-\epsilon)\frac{m}{2}\log\left(1+\frac{a_n\beta_n}{\frac{\sigma^2}{P}+(m-1)\mu_\gamma}\right) \tag{66}$$

which we want to compare to the result in [3]:

$$T=(1-\epsilon_n)\alpha k\frac{\log(nQ_n(\beta_n))}{\log n}\times\log(1+\frac{a_n\cdot\beta_n}{\frac{\sigma^2}{P}+(k-1)\mu_\gamma}) \tag{7}$$

An important advantage of the scheme presented in our work is that the throughput doesn't depend on *k* (maximum number of simultaneous transmissions determined by the algorithm in [4]), but on *m*, which is to be chosen by the service provider. Thus our model is more flexible, and the quality of service can be directly improved through an investment in access points.

If we take *m*=*k*, the gain that the addition of infrastructure nodes gives us is:

$$G=\frac{\log n}{2\alpha\log(np)} \tag{67}$$

which is positive whenever

$$\log(p)<\frac{1-2\alpha}{2\alpha}\log(n) \tag{68}$$

This gain depends on *α*, which is a parameter introduced in [3] and that depends on the performance of the algorithm to find the vertex-disjoint paths in a certain realization of it, and is just supposed to be bigger than one.

Section IX. Conclusions and future work

The aim of this work was to find an analytical expression of the achievable throughput in an ad hoc network when adding a number of fixed nodes (access points or infrastructure nodes) connected by high capacity wired links.

We have proposed three different scheduling schemes with that purpose and seen that, by applying them to our network model, one can improve its throughput. We have also stated that, to maximize this improvement, the usage of vertex-disjoint paths as in [3] is very sub-optimal. Thus, we have analyzed the possible network performance when one rules out this scheduling scheme in order to increment the usage of the wired nodes. The advantages of it are the reduction of the number of simultaneous interfering transmissions, the possibility of scheduling a higher number of simultaneous transmissions and the promotion of the usage of the wired high-capacity links. Sections V and VI are devoted to introduce two possible scheduling solutions that do not use vertex-disjoint paths, but that are based on the consideration of the network as a set of rings and levels around the infrastructure. In Section V, we avoided collisions by allowing just one transmission in every sub-graph and Section VI raised the possibility of scheduling many simultaneous transmissions in a sub-graph at a time.

We have also determined that, in the schemes of Sections V and VI, the main influence of the number of access points *m* in the performance resides in the effect that it has in the network topology, which determines the circles' and levels' construction. This means that using any of the schemes proposed in Sections V and VI, one will have a similar performance basically influenced by the average distance in terms of hops that separate any node from the closest infrastructure nodes.

According to this, we have found an expression of the minimum connectivity, given a number of nodes and a number of access points, that one needs to assure that the average number of hops is minimum, i. e. $\tilde{h}=1$. Complementary, we have also found an expression of the minimum number of access points that one needs, given a network (i. e. given a pair of *n* and *p*), to have, with high probability, just one level.

The work performed in this thesis is destined to be published as a paper, whose work is still in progress, in IEEE Wireless Communications Magazine. The Center for Communications and Signal Processing Research (CCSPR) at New Jersey Institute of Technology (NJIT), where this work was developed, is currently simulating the presented results.

The results here put forward are open and dependent, in every context, of the probability distribution of the connection strengths. Thus, they might be highly variable depending on the environment. The study of concrete probability distribution functions of the strengths remains to be done and is going to be developed in the CCSPR of NJIT shortly.

Also, as pointed above, the schemes here presented introduce an improvement when the number of access points is enough to guarantee that the network topology has just one level. Future study lines might be opened to find new scheduling schemes that improve the results obtained when this condition cannot be accomplished.

Appendices

Appendix I

Claim: The throughput in (13) is always increasing with m if $\frac{\sigma^2}{P}-\mu_\gamma \geqslant 0$

Proof:

From (13) we state that he optimum *k* is the one that maximizes

$$t=k\cdot\log\left(1+\frac{a_n\beta_n}{\frac{\sigma^2}{P}+(k-1)\mu_\gamma}\right)$$

To find this maximum, we need to find some value of *k* where the derivative is 0. However, if the derivative is always positive, the optimum *k* is the biggest possible.

The derivative is:

$$\frac{\partial t}{\partial m}=\log\left(1+\frac{a_n\beta_n}{\frac{\sigma^2}{P}+(k-1)\mu_\gamma}\right)-\frac{k}{1+\frac{a_n\beta_n}{\frac{\sigma^2}{P}+(k-1)\mu_\gamma}}\frac{a_n\beta_n\mu_\gamma}{\left(\frac{\sigma^2}{P}+(k-1)\mu_\gamma\right)^2}$$

Recall that *m* and $\tilde{h}$ are values bigger than 1 and $0<a_n,\beta_n<1$, thus $m\tilde{h}>1>a_n\beta_n$.

Therefore, if $\frac{\sigma^2}{P}-\mu_\gamma \geqslant 0$:

$$\frac{\sigma^2}{P}+(m\tilde{h}-1)\mu_\gamma=\frac{\sigma^2}{P}-\mu_\gamma+m\tilde{h}\mu_\gamma>m\tilde{h}\mu_\gamma>a_nb_n\mu_\gamma>0$$

Using these inequalities we can prove that every factor in the derivative above is positive.

Since $a_n,b_n>0$ and $\frac{\sigma^2}{P}+(m\tilde{h}-1)\mu_\gamma>0$, $\left(1+\frac{a_nb_n}{\frac{\sigma^2}{P}+(m\tilde{h}-1)\mu_\gamma}\right)>1$ and thus:

$$\left(1+\frac{a_n b_n}{\frac{\sigma^2}{P}+(m\tilde{h}-1)\mu_\gamma}\right)^m>1 \quad \text{and} \quad \ln\left(1+\frac{a_n b_n}{\frac{\sigma^2}{P}+(m\tilde{h}-1)\mu_\gamma}\right)>0 \quad .$$

On the other hand, since $\frac{\sigma^2}{P}+(m\tilde{h}-1)\mu_\gamma>a_n b_n \mu_\gamma$, $\frac{\sigma^2}{P}+(m\tilde{h}-1)\mu_\gamma-a_n b_n \mu_\gamma>0$ and

therefore $\dfrac{\frac{\sigma^2}{P}+(m\tilde{h}-1)\mu_\gamma-a_n b_n \mu_\gamma}{\left(\frac{\sigma^2}{P}+(m\tilde{h}-1)\mu_\gamma\right)^2}>0$.

Appendix II

Claim: The number of simultaneous hops in Table 4 is always bigger than *m*, if *n>100*.

Proof:

Recall from Table 4 that the number of simultaneous hops in a circle is

$$h_i=(-1)^i m+m\sum_{j=1}^{i}(np)^j(-1)^{i-j}=(-1)^i m\left[1+\sum_{j=1}^{i}(-np)^j\right]=(-1)^i m\left[1+\frac{(-np)^{i+1}-(-np)}{(-np)-1}\right]$$

where we used $(-1)^{-j}=(-1)^j$.

$$h_i=(-1)^i m\left[1+np\frac{(-np)^i-1}{np+1}\right]$$

This means that h_i is bigger than *m* if and only if $(-1)^i\left[1+np\frac{(-np)^i-1}{np+1}\right]>1$

If *i* is even: $(-1)^i\left[1+np\frac{(-np)^i-1}{np+1}\right]=1+np\frac{(np)^i-1}{np+1}>1$ since *np>1* and thus $(np)^i-1>0$.

And if *i* is odd: $(-1)^i\left[1+np\frac{(-np)^i-1}{np+1}\right]=-1\left[1+np\frac{-(np)^i-1}{np+1}\right]$ which is bigger than 1 if and only if $1+np\frac{-(np)^i-1}{np+1}=1-np\frac{(np)^i+1}{np+1}<-1$ if and only if $np\frac{(np)^i+1}{np+1}>2$.

Since np>1 then, $\frac{(np)^i+1}{np+1}>1$ hence must have np≥2, but since np>log n, we must have log n>2 or n>100.

Appendix III

Claim: The throughput in (34) is always increasing with m if $\frac{\sigma^2}{P}-\mu_\gamma \geqslant 0$

Proof:

From (34) the throughput is increasing if the following term is increasing with *m*

$$m \cdot \log\left(1+\frac{a_n \beta_n}{\frac{\sigma^2}{P}+(m\tilde{h}-1)\mu_\gamma}\right)$$

But, since the logarithm is a monotone function, it is enough to show that,

$$t=\left(1+\frac{a_n \beta_n}{\frac{\sigma^2}{P}+(m\tilde{h}-1)\mu_\gamma}\right)^m$$

is increasing with *m*, or equivalently if and only if the derivative with *m* is always positive.

The derivative is:

$$\frac{\partial t}{\partial m}=\ln\left(1+\frac{a_n \beta_n}{\frac{\sigma^2}{P}+(m\tilde{h}-1)\mu_\gamma}\right)\cdot\left(1+\frac{a_n \beta_n}{\frac{\sigma^2}{P}+(m\tilde{h}-1)\mu_\gamma}\right)^m\cdot\left(\frac{\frac{\sigma^2}{P}+(m\tilde{h}-1)\mu_\gamma-a_n\beta_n\tilde{h}\mu_\gamma}{\left(\frac{\sigma^2}{P}+(m\tilde{h}-1)\mu_\gamma\right)^2}\right)$$

Recall that *m* and $\tilde{h}$ are values bigger than 1 and $0<a_n, \beta_n<1$, thus $m\tilde{h}>1>a_n\beta_n$.

Therefore, if $\frac{\sigma^2}{P}-\mu_\gamma \geqslant 0$:

$$\frac{\sigma^2}{P}+(m\tilde{h}-1)\mu_\gamma=\frac{\sigma^2}{P}-\mu_\gamma+m\tilde{h}\mu_\gamma>m\tilde{h}\mu_\gamma>a_n b_n\mu_\gamma>0$$

Using these inequalities one can prove that every factor in the derivative above is positive.

In fact $a_n, b_n>0$ and $\frac{\sigma^2}{P}+(m\tilde{h}-1)\mu_\gamma>0$, $\left(1+\frac{a_n b_n}{\frac{\sigma^2}{P}+(m\tilde{h}-1)\mu_\gamma}\right)>1$ and thus:

$$\left(1+\frac{a_n b_n}{\frac{\sigma^2}{P}+(m\tilde{h}-1)\mu_\gamma}\right)^m>1 \quad \text{and} \quad \ln\left(1+\frac{a_n b_n}{\frac{\sigma^2}{P}+(m\tilde{h}-1)\mu_\gamma}\right)>0 \quad .$$

Also, since $\frac{\sigma^2}{P}+(m\tilde{h}-1)\mu_\gamma>a_n b_n \mu_\gamma$, $\frac{\sigma^2}{P}+(m\tilde{h}-1)\mu_\gamma-a_n b_n \mu_\gamma>0$ and therefore

$$\frac{\frac{\sigma^2}{P}+(m\tilde{h}-1)\mu_\gamma-a_n b_n \mu_\gamma}{\left(\frac{\sigma^2}{P}+(m\tilde{h}-1)\mu_\gamma\right)^2}>0 \quad .$$

Appendix IV

The binomial distribution *B(n,p)* is a discrete probability distribution of the number of successes in a sequence of *n* independent experiments where every experiment has a probability of success *p*, which is also called Bernoulli experiment.

The probability mass function of the binomial distribution is:

$$P(K=k)=\binom{n}{k}p^k(1-p)^k$$

The mean and the variance of the binomial distribution are proved to be:

$$E(x)=np$$

$$\sigma_n^2=np(1-p)$$

In our work, we use this distribution and apply its properties to define the behavior of the number of nodes in every ring, considering that the existence of a good connection is a success in a set of experiments.

Appendix V

Claim: $n-n_1<1$ if $p<1-\sqrt[m]{\frac{5n-\sqrt{(5n)^2-16n}}{8n}}$.

Proof:

Recall from (49) that

$$n-n_1<5n(1-p)^m-4n(1-p)^{2m} \quad .$$

Define $x=(1-p)^m$, so that we can express $n-n_1<5nx-4nx^2$. If $5nx-4nx^2<1$, the number of nodes not included in the first level $n-n_1$ is smaller than 1.

Solving the second order equation $4nx^2-5nx+1=0$ we get two possible limits for *x*:

$x_1=\frac{5n+\sqrt{(5n)^2-16n}}{8n}$ and $x_2=\frac{5n-\sqrt{(5n)^2-16n}}{8n}$. Since we expect *n* to be big, we can approximate $x_1\approx\frac{5n+\sqrt{(5n)^2}}{8n}=\frac{10n}{8n}>1$ which isn't a possible solution for $(1-p)^m$ since $1-p<1$.

Therefore, we have found that the limit for *x* is $x_2=\frac{5n-\sqrt{(5n)^2-16n}}{8n}$ which means $(1-p)^m>\frac{5n-\sqrt{(5n)^2-16n}}{8n}$ and thus $p<1-\sqrt[m]{\frac{5n-\sqrt{(5n)^2-16n}}{8n}}$ Q. E. D.

Appendix VI

Claim: $n-n_1<1$ if $m>\dfrac{\log\left(\dfrac{5n-\sqrt{(5n)^2-16n}}{8n}\right)}{\log(1-p)}$.

Proof:

Recall from (49) that

$$n-n_1<5n(1-p)^m-4n(1-p)^{2m}\quad .$$

Define $x=(1-p)^m$, so that we can express $n-n_1<5nx-4nx^2$. If $5nx-4nx^2<1$, the number of nodes not included in the first level $n-n_1$ is smaller than 1.

Solving the second order equation $4nx^2-5nx+1=0$ we get two possible limits for *x*:

$x_1=\dfrac{5n+\sqrt{(5n)^2-16n}}{8n}$ and $x_2=\dfrac{5n-\sqrt{(5n)^2-16n}}{8n}$. Since we expect *n* to be big, we can approximate $x_1\approx\dfrac{5n+\sqrt{(5n)^2}}{8n}=\dfrac{10n}{8n}>1$ which isn't a possible solution for $(1-p)^m$ since $1-p<1$.

Therefore, we have found that the limit for *x* is $x_2=\dfrac{5n-\sqrt{(5n)^2-16n}}{8n}$ which means $(1-p)^m>\dfrac{5n-\sqrt{(5n)^2-16n}}{8n}$ and thus $\log(1-p)^m>\log\left(\dfrac{5n-\sqrt{(5n)^2-16n}}{8n}\right)$ and

$$m>\frac{\log\left(\dfrac{5n-\sqrt{(5n)^2-16n}}{8n}\right)}{\log(1-p)}$$ Q. E. D.

References

[1] O. Dousse, P. Thiran and M. Hasler, “Connectivity in ad-hoc and hybrid networks,” *IEEE Inforcom 2002*, pp 1079-1088.

[2] P. Gupta and P. R. Kumar, “The capacity of wireless networks,” *IEEE Transactions on Information Theory*, vol. 46, no. 3, pp. 388–404, Mar. 2000.

[3] R.Gowaikar, B. Hochwald and B. Hasabi, “Communication Over a Wireless Network With Random Connections,” *IEEE Transactions on Information Theory,* vol. 52, no. 7 pp. 2857-2871, Jul. 2006.

[4]A. Z. Broder, A. M. Frieze, S. Suent and E. Upfals, “An Efficient Algorithm for the Vertex-Disjoint Paths Problem in Random Graphs”, *Proc. 7th Symp. Discrete Algorithms*, pp. 261-268, 1996.

[5] B. Bollobás, “Random Graphs”, *Cambridge University Press*, 2nd edition, Jan. 2001.

[6] B. Bollobás, “Modern Graph Theory” *Springer*, 2nd edition, Au. 2002.

[7] S. Janson, T. Luczak and A. Rucinski, “Random Graphs”, *Wiley-Interscience*, 2000.

[8] F. Chung and L. Lu, “The diameter of random sparse graphs,” *Adv. Appl. Math.*, vol. 26, pp. 257–279, 2001.