NPART - Node Placement Algorithm for Realistic Topologies in Wireless Multihop Network Simulation

Bratislav Milic and Miroslaw Malek

Institute for Informatics
Humboldt Universität zu Berlin,
Berlin, Germany
(milic,malek)@informatik.hu-berlin.de

Abstract. Despite a considerable number of topology generation algorithms for simulation of wireless multihop networks it is difficult to find one with output similar to real networks [13].
In this paper, we propose NPART – a Node Placement Algorithm for Realistic Topologies whose created topologies resemble networks encountered in reality. The algorithm is flexible since it is sufficient to provide it with different input data to obtain different topologies. To demonstrate its quality and adaptivity, we compare topologies created by PART algorithm with topology samples from open wireless multihop networks in Berlin and Leipzig. Compared with real topologies, the generated topologies have almost identical node degree distribution, similar number of cut edges and vertices, and distribution of component sizes after bridge removal.
The importance of node placement algorithm is demonstrated by comparing ns-2 simulation results for grid and uniform node placement with NPART generated topologies. Simulation results show that quality of node placement model plays as important role in simulation outcome as the accuracy of wireless signal propagation model.
To our best knowledge, this is the first node placement algorithm for wireless multihop networks capable of creating topologies that have properties observed in user initiated networks.

1 Introduction

Wireless multihop networks (WMNs) are used for various purposes such as Internet access, sensing applications, and military usage. Different applications require dedicated protocols in order to offer quality and performance to end users in presence of limited resources. Verification of developed protocols is necessary and usually it is performed by simulators, due to their low operating cost and fast setup. Quality of simulation directly depends on simulation model. Model of WMNs is complex and it consists of five sub-models: node model describes hardware and software of a node; deployment and mobility models provide node positions and their movement patterns; radio model describes the characteristics of the radio used by the node; wireless signal propagation model deals with attenuation and characteristics of wireless channel; traffic models define traffic patterns in a network.

Some of sub-models are based on real data measurements (e.g., wireless signal propagation [1], traffic models [17]). However, some of them are synthetic and somewhat arbitrary like the topology generators/node placement models.

In [11], [12] and [13] we have presented results of measurements from wireless mesh networks in Berlin and Leipzig. Approximately 1500 topological samples have been taken from each network. The analysis showed that properties of artificial topologies are substantially different from properties observed in reality. Table 1 summarizes differences observed for average node degree, network diameter, number of biconnected components and articulation points.

The main approach in natural sciences such as physics or biology is to observe reality and create a model that reflects it. In computer science a frequent activity is to create artifacts

	Average Nodes	Average degree	#Biconnected components	Network Diameter	Articulation points
Berlin	315	4.02	99.22	20.52	75.93
Leipzig	586	4.35	120.1	23.69	93.32
Uniform	400	5.31	30.6	37.76	32.46
RWM	400	7.6	31.7	25.15	21.22
20x20 Grid	400	3.8	1	38	0

Table 1. Comparison of real and synthetic topologies.

and develop unrealistic models that do not reflect the reality. And so is with most of the node placement models: they have not been inspired by reality nor verified by the measurements. In order to correct this fundamental issue, we propose algorithm that produces realistic topologies.

The paper is organized as follows: Section 3 reviews existing topology generators for WMN simulations. In Section 4, we present our node placement/topology creation algorithm. Section 5 shows that the developed algorithm generates topologies whose properties reflect the properties of real topologies.

2 Definitions

We use undirected graphs to model communication in WMNs. In communication graph, network nodes are represented as vertices. If node p is able to communicate with node q there exists edge pq in the graph.

Definition 1. *Two vertices a and b are adjacent if there exists an edge between them ($ab \in E(G)$). If vertex a belongs to an edge e, e and a are incident.[18]*

Definition 2. *A walk of length k is a sequence $v_0, e_1, e_2, ..., e_k, v_k$ of vertices and edges such that $e_i = v_{i-1}v_i$ for all i. A trail is a walk with no repeated edge. A path is a walk with no repeated vertex.*

Definition 3. *If in graph G exists a path between vertices u and v, the distance $d(u, v)$ between u and v is the number of edges on path (u, v). The diameter of graph G is $max_{(u,v)\in V(G)} d(u, v)$.*

Definition 4. *The degree of a vertex v in a graph G, written $d_G(v)$ or $d(v)$ is the number of edges incident on v. A pendant vertex is a vertex of degree 1.*

Definition 5. *Maximally connected subgraph is a graph such that there exists a path between any pair of nodes (p, q). Components of a graph G are its maximally connected subgraphs. A bridge in a graph (cut-edge) is an edge whose deletion increases the number of components. An articulation point in a graph (cut-vertex) is a vertex whose deletion increases the number of components in the graph.*

Definition 6. *Given an undirected graph, a degree sequence is a monotonic nonincreasing sequence of the vertex degrees (valencies) of its graph vertices. A degree set is a set of integers that make up a degree sequence.*

Definition 7. *The frequency of an event i is the number n_i of times the event occurred in the experiment. The frequency can be absolute, when the counts n_i are given and relative, when counts are normalized by the total number of events.*

3 Related Work

Calvert et al. [4] propose two algorithms that support locality and hierarchy properties observed in the Internet for creation of topologies of wired networks that resemble typical internetworks. Zegura et al. [19] compare these algorithms with other topology generators and demonstrate their impact on simulation results on example of multicast routing. Unfortunately, the experiences from generation of wired network topologies cannot be applied in wireless context, since in wired networks there does not exist strong dependency between node location, internode distance and existence of links, like in wireless networks.

In WMN research, the most frequent node placement models are uniform, chain and grid. In uniform placement model a placement area (rectangular or circular) of size $|A|$ is chosen and n nodes are placed inside of it with uniform probability $p_{uniform} = \frac{n}{|A|}$. If placement area is rectangular $((0, x_{max}), (0, y_{max}))$, this is typically achieved by sampling x coordinate of a vertex from $U(0, x_{max})$ and y from $U(0, y_{max})$. Chain placement model places nodes on a line on equal distance. In grid placement model, nodes are located at intersections of a rectangular grid. Number of edges in grid depends on node communication radius, shape and size of cell. In the literature these parameters are typically chosen so that all nodes that are not on the grid border have degree of four.

It is particularly difficult to create connected low-density topologies with existing randomized models. To ensure connectivity of simulated network, the average node degree is increased. Bettstetter shows in [2] that for uniform placement process, it is required that nodes have average degree of 10.8 if we are to have network that is connected with probability of 0.99. Li et al. [9] provide even higher estimation – they claim that obtaining of the same connectivity probability requires 13.78 neighbor nodes on the average. Such dense networks have strong impact on simulation results since:

- Network diameter is significantly reduced
- Numerous independent paths between each pair of nodes exist
- Failure of individual nodes does not impact the connectivity nor the functionality of the network

The need for improved node placement model in WMNs has been already noticed and several non-homogeneous models have been proposed. Bettstetter et al. [3] place nodes in accordance with the uniform homogeneous process and then apply thinning to it. The thinning operation removes nodes from a network that have less than k neighbors within radius r (denoted as tr in [3]). Parameters k and r are specified by the user and they control the level of inhomogeneity of the topology. Bettstetter et al. also calculate several statistics for the obtained model, such as probability of nearest neighbor survival and distance to closest neighbor. However, they discuss only the node placement, ignoring the properties and connectivity of topologies that can be obtained from it.

Liu and Haenggi [10] propose two quasiregular placement models. In first, vertex coordinates are Gaussian distributed with the mean given by regular grid points. The second selects vertices from a uniform placement model such that every selected vertex is closest to a regular grid point. The obtained topologies resemble the grid structure but they are not as regular as grids. Various other variations and inhomogeneity models exist: in [6], two-dimensional Gaussian distribution is used to determine location of sensor nodes. This idea can be extended so that there are multiple (two or more) vertex focal points, each of the points having a non-uniform distribution attached to it.

Onat and Stojmenovic [14] propose considerably different approach. They have developed several algorithms that create connected topologies with high probability and allow user to choose the average node degree. The shape of topologies primarily depends on the selected algorithm. The algorithms do not guarantee connectivity of their output - if the end result is not connected, algorithm is restarted. Their analysis focuses on algorithm complexity and probability that created graph is connected. The probability density function for node degree for each of algorithms is presented and the differences among placement topologies created by different algorithms is informally (visually) demonstrated.

4 NPART – a Node Placement Algorithm for Realistic Topologies

Our goal is to develop a node placement / topology generating algorithm that is

- **Adaptive** – it is capable to create more than one node distribution type.
- **Realistic** – if algorithm receives input based on measurements from a real network, the topologies that it produces should have similar properties as the original networks.
- **Random** – the algorithm does not merely re-create a sampled topology from measured node locations, wireless device parameters (power, receiving threshold), signal to noise ratio. It is capable to create new, random topologies but preserving the properties of adaptivity and reality.

The starting point for algorithm creation is taken from [12]. The following sociological and technological reasons that shape topologies of real networks are identified:

- It is more likely that new participants join the network in areas where connectivity is already good.
- A participant in the network expects to have at least a single communication link to the remainder of the network, possibly creating large number of pendant nodes.
- A pendant node might become a seed for a new, larger and well connected subnetwork.
- It is the network that specifies the area it occupies, not the other way around. So, instead of defining the node placement area like in most of the existing placement algorithms, the network should be allowed to grow.

4.1 Algorithm description

The algorithm is presented in Figure 1. As input parameters, algorithm accepts n, the number of vertices to be placed and communication radius r. The algorithm calculates topologies based on the path-loss model. The user should specify the additional propagation models in the simulation (shadowing and Rayleigh fading [1]) to create realistic simulation results. If needed, users can also customize the algorithm by specifying appropriate metrics that take propagation models in account (e.g., number of edges in the graph with expected packet loss higher than 0.5 in presence of shadowing on the wireless channel).

In first iteration of the algorithm the first vertex is placed at an arbitrary point (x,y) in two-dimensional space. The variables $minX$ and $maxX$ are initialized to x, $minY$ and $maxY$ to y. Values of these variables from iteration I_k are used to determine the placement area of nodes in the next iteration: in I_{k+1} x coordinate of candidate nodes is uniformly sampled from ($minX - r$, $maxX + r$), while y coordinate is chosen from ($minY - r$, $maxY + r$). This enables the network to grow, without need to predetermine its geographical size.

The algorithm consists of three loops. The innermost ensures that the candidate vertex is connected with the remainder of the graph, since it is possible that a vertex placed in rectangle (($minX - r$, $minY - r$), ($maxX + r$, $maxY + r$)) is not connected to already placed vertices. For instance, in iteration I_4 Vertex 0 in Figure 2 is disconnected from nodes placed in I_3. The Vertex 0 is ignored and new candidate vertex is generated.

Once the candidate vertex forms the connected topology with the existing graph, a user-defined metric is applied to it. Section 4.2 describes four metrics that we have implemented and tested. If the candidate vertex has lower metric than previous candidates, it is stored as the

```
place nodes(nodes n, comm.radius r,
candidates to evaluate in iteration retries):
   placedNodes = place first node arbitrarily at (x,y)
   minX=maxX=x
   minY=maxY=y
   repeat
      minMetric=∞, candidateN = null
      repeat
         repeat
            x-coordinate=U(minX-r, maxX+r)
            y-coordinate=U(minY-r, maxY+r)
            create node candidateN from coordinates
         until (candidateN ∪ placedNodes is connected)
         m=apply metric on placedNodes ∪ candidateN
         if(m < minMetric)
            bestCandidate = candidateN
            minMetric = m
         endif
      until(retries candidates evaluated)
      update minX, maxX, minY, maxY based on
bestCandidate location
      placedNodes = placedNodes ∪ bestCandidate
   until(all n nodes placed)
```

Fig. 1. NPART pseudo code description.

bestCandidate and minimal metric value is updated. After evaluation of *retries* connected candidates, the best candidate is added to the topology and $minX$, $maxX$, $minY$, $maxY$ variables are updated, if needed. For example, if Vertex 3 is the best candidate out of three candidate nodes in Figure 2, variables $maxX$ and $minY$ must be updated. Number of evaluated candidates *retries* is parameter of the algorithm, and as the number of evaluated candidates grows, increases the chance that the produced topology is closer to the predefined goal.

After placement of all n nodes their locations can be shifted so that they are in rectangle $((0,0), (|maxX - minX|, |maxY - minY|))$. This step is optional and does not influence the functionality of the algorithm.

4.2 Quality metrics

Metric that evaluates quality of topology candidates is as important as the algorithm itself. Inappropriate metric results in unsatisfactory topologies. Unfortunately, there does not exist universal metric. User must define them and perform tests to check whether the algorithm and metric produce desired topologies.

Our goal are topologies that have properties observed in real, user-initiated networks. In process of selecting what should be the input for metric, it was obvious from experience with existing placement models that generic parameters such as node density or average node degree cannot capture desired level of detail. Realistic topologies can be produced only with input parameters that originate from measurements.

Capturing spatial node distribution and link quality metrics (e.g., signal to noise ratio, bit error ratio, packet loss probability) in real network would be an excellent input for a vertex place-

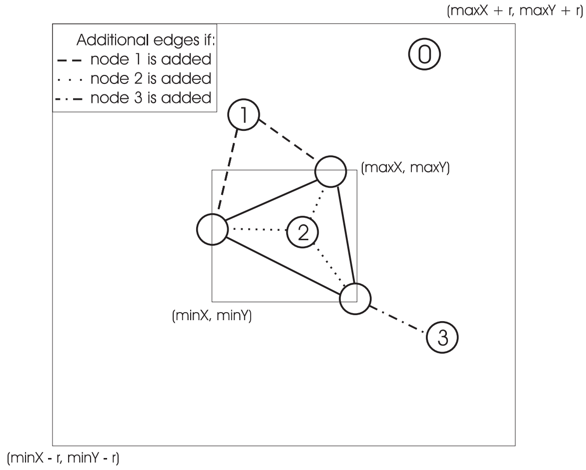


Fig. 2. Placement area and candidate nodes.

ment algorithm, but impossible to implement (e.g., due to privacy concerns not all participants in a real network are willing to disclose their geographical locations). Additionally, quality of antennas cannot be automatically collected, signal propagation environment is heterogenous and its impact on link quality cannot be accurately measured with of-the-shelf components that are commonly used. In rare cases when it is possible to take samples from user initiated networks, typically only the topological information is available, without node location data.

We have implemented several metrics that use node degree frequencies (Definition 7). The degree frequency is a compromise between detail level, data anonymity and feasibility of sampling. It is easily extracted from networks, regardless of the routing protocol type (proactive or reactive) and it is anonymous by its definition. In proactive protocols, it is trivial to calculate it. In case of reactive routing protocols where no global topology view exists, node degrees can be easily obtained assuming that nodes in the network are cooperative: each node samples its degree and shares it with the central repository. Additionally, small errors in sampling (a node falsely reports its degree, or topology has not converged to its steady state) are hidden by larger set of correct data.

The implemented metrics are shown in Figure 3. As input for *distance* and *adaptive* metrics, we calculate relative node degree frequency from degree sequences from all samples that were taken from Berlin's and Leipzig's networks. The relative node degree frequency of real network is multiplied by number of nodes that should be placed by the algorithm, creating absolute vertex degree frequency for target topology *target*. For each candidate vertex that is evaluated, absolute degree frequency *candidate* of topology that it creates with already placed nodes is calculated and compared with the target frequency.

The simplest, *distance* metric is a variation of the Manhattan metric:

$$\sum_{degrees}^{d} (1_{target_d - candidate_d > 0} \cdot (target_d - candidate_d) + 1_{target_d - candidate_d < 0} \cdot p \cdot (candidate_d - target_d)) \tag{1}$$

where $1_A(x)$ is indicator function, returning one if $x \in A$, zero otherwise. The metric sums difference between proposed and target vertex frequency if difference is positive. If it is negative (produced topology has more nodes of certain degree than the target topology), absolute value of difference is multiplied with penalty factor p. The penalty factor reflects user's tolerance for

distance metric(target frequency *target*, candidate frequency *candidate*, penalty p):
$metric = 0$;
for (i in degrees of *target*)
if(target[i]-candidate[i]<0)
$metric = metric + |target[i] - candidate[i]| \cdot p$
else $metric = metric + target[i] - candidate[i]$
return *metric*

adaptive metric(target frequency *target*, candidate frequency *candidate*, frequency of already placed *placed*, penalty p):
$weights = normalize\_to\_one(|target - placed|)$
$metric = 0$;
for (i in degrees of *target*)
if(target[i]-candidate[i] < 0)
$metric = metric + |target[i] - candidate[i]| \cdot p$
else $metric = metric + (target[i] - candidate[i]) \cdot weights[i]$
return *metric*

secondaryDistribution(canidate graph topology *topology*, secondary degree relative frequency *secondary_target*)
Initiate list of empty absolute degree frequencies *list*
for each node n in *topology*
select frequency based on current node degree
$current\_freq = list[degree(n)]$
for each neighbor ng of n
$current\_freq[degree(ng)] + +$
endfor
list=calculate relative frequencies from absolute frequencies
$metric = 0$
for i in 1:length(secondary_target)
$metric = metric + |secondary\_target[i] - list[i]|$
return *metric*

combined(target frequency *target*, candidate frequency *candidate*, frequency of already placed *placed*, penalty p, secondary weight s, secondary degree relative frequency *secondary_target*, candidate graph topology *topology*,)
return **adaptive**(target, candidate, placed, p) + **secondary**(target,topology) $\cdot s$

quality(target frequency *target*, generated frequency *generated*)
$quality = 0$
for (i in degrees of *target*)
$quality = quality + |target[i] - generated[i]|$
return *quality*

Fig. 3. Implemented metrics.

Degrees	1	2	3	4	5	Distance metric	Adaptive metric
Absolute Target degree frequency	2	5	3	2	1	0	0
Absolute Placed degree frequency	0	3	0	0	0	10	2
Weights w_d	0.2	0.2	0.3	0.2	0.1		
Candidate 1	0	2	2	0	0	9	1.8
Candidate 2	0	0	4	0	0	15	6.9
Candidate 3	1	2	1	0	0	9	1.9

Table 2. Metric example for node candidates in Figure 2. Parameter p is set to five.

overloading of degrees: with decrease in tolerance, user increases the factor p. If $p = 1$, the *distance* metric is identical to the Manhattan metric.

The drawback of the *distance* metric is its impossibility to detect stronger need for creation of vertices with certain degree. Some degrees are more frequent in target degree frequency so topologies that produce them should be rewarded. For instance, if the algorithm should create 20 vertices with degree two and three vertices with degree four, the metric should give greater reward (smaller metric value) in early iterations of the algorithm to topologies that increases number of vertices with degree two. The *adaptive* metric resolves this issue:

$$\sum_{degrees}^{d} (1_{target_d - candidate_d > 0} \cdot (target_d - candidate_d) \cdot w_d + 1_{target_d - candidate_d < 0} \cdot p \cdot (candidate_d - target_d)) \quad (2)$$

and

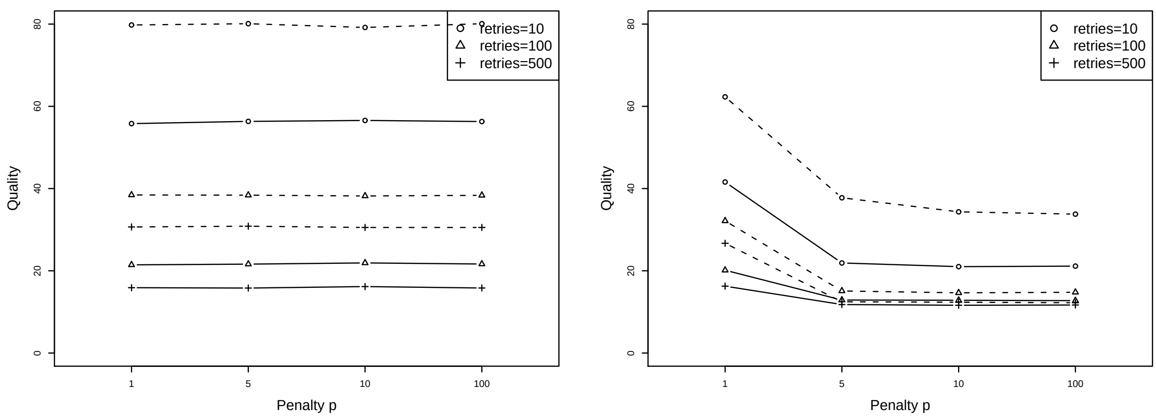
$$w_d = \frac{|target_d - placed_d|}{\sum_{degrees}^{d} |target_d - placed_d|}$$

where *placed* is the absolute degree frequency of vertices that are already placed.

Figure 2 and Table 2 illustrate use of *distance* and *adaptive* metrics. Three iterations of algorithm have been executed and in fourth iteration three candidates are evaluated. The *distance* metric calculates equal value for Candidates 1 and 3, so either of them can be selected as the best candidate. Adaptive metric correctly chooses Candidate 1 as better, since it satisfies greater need to create node of degree three (after I_3 three more nodes with degree three are required) than to create node of degree one like the candidate Vertex 3 (after I_3 two more nodes of degree one are needed to reach the target absolute degree frequency).

Figure 4 shows behavior of algorithm, used with *distance* and *adaptive* metrics, for different combinations of parameters p and *retries*. Input are degree frequencies from Freifunk Berlin and Freifunk Leipzig networks. As quality measure of algorithm we use Manhattan metric between targeted and produced absolute degree frequencies. The topologies produced with the *adaptive* metric are considerably better than the topologies produced with the *distance* metric. The *distance* metric is invariant of the penalty parameter. The *adaptive* metric is sensitive to the penalty parameter, but if penalty is equal to or larger than five the quality of produced topologies stabilizes. As expected, increase in number of retries improves quality of produced topologies for both metrics. However, there is no significant quality improvement if number of retries is increased from 100 to 500. Based on this evaluation, we conclude that *adaptive* metric is better, and it will be analyzed in more detail in Section 5.

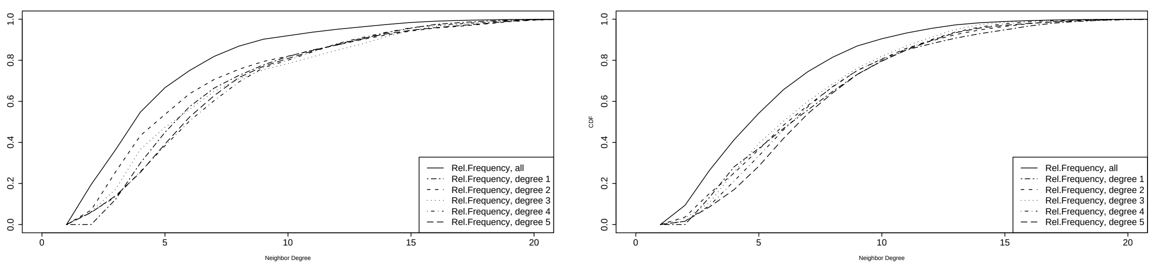
It is possible to extract additional degree data from topologies. Let us observe relative degree frequency of neighbors of a node p under condition that degree of p is k. As Figure 5 shows, the



(a) Distance metric.

(b) Adaptive metric.

Fig. 4. Average quality of produced topologies for *distance* and *adaptive* metrics. Solid line represents Leipzig distribution while dashed is for distribution from Berlin's network.



(a) Berlin.

(b) Leipzig.

Fig. 5. Conditional degree distributions.

conditional relative frequencies differ considerably among themselves and to the joint relative frequency.

The metric $secondaryDistribution$ uses these differences. First it calculates set of conditional relative degree frequencies for candidate topology and then compares them (using Manhattan metric) with the target conditional relative degree frequency.

The *combined* metric is a linear combination of *adaptive* and $secondaryDistribution$ metrics. It is possible to vary the penalty factor p in the *adaptive* metric and weight s for the $secondaryDistribution$ metric.

5 Evaluation of Created Topologies

This section compares properties of topologies produced by the NPART algorithm and presented metrics with properties of real networks. The quality of the algorithm is demonstrated and different combinations of algorithm parameters are tested to find the one that produces best results.

The data that is used to provide input degree distributions for the algorithm and for later comparison is taken from Berlin and Leipzig networks. Sampling methodology, detailed analysis

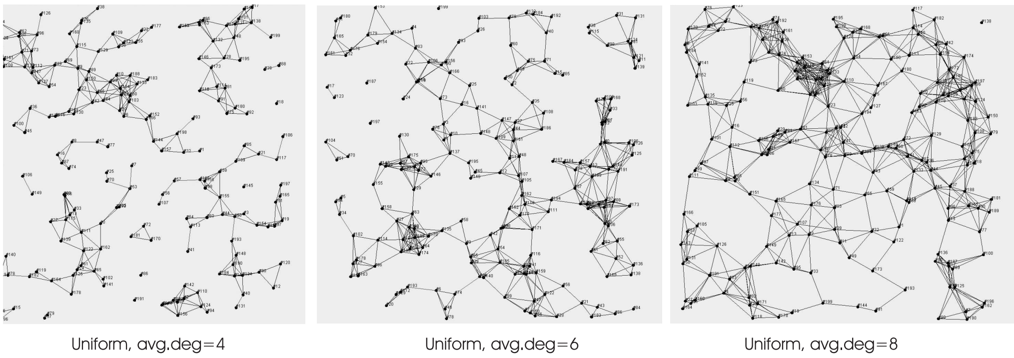


Fig. 6. Visual comparison of topologies created by uniform placement model.

and comparison with synthetic placement models are presented in [13]. There is a small change in datasets if compared with [12] and [13]: only the main partition (largest maximally connected subnetwork) is considered for algorithm's degree data input and later result comparison. The main partition in Berlin has 275 and in Leipzig 346 nodes on the average.

To illustrate the improvements brought by the NPART, it is compared with the ubiquitous uniform placement model. Uniform placement algorithm is set to create topologies with the average node degree of six. The average node degree is substantially lower than proposed in [2] and [9] for networks connected with high probability so it is possible that such a graph is partitioned. Increasing the average node degree above six improves connectivity but creates even greater discrepancy with measurement results (e.g. bridges do not exist in generated topologies), while decreasing it creates highly partitioned graphs (Figure 6). Since the size of Berlin's and Leipzig's main partition differs there are also two uniform placement scenarios, with 275 and 346 nodes. The data for comparison is collected in 500 executions of each scenario.

The proposed placement algorithm is also run with two basic setups: 275 vertices and degree data input from Berlin's network (NPART/Berlin), and 346 vertices and data input from Leipzig's network (NPART/Leipzig). The parameter *retry* is set to 150 while parameters for penalty p and secondary metric weight are varied to take values from set $\{0, 1, 5\} \times \{0, 1, 5\}$. Not all results are presented since some parameter combinations do not create reasonable results: as soon as *secondary* weight is higher than the penalty p, the algorithm becomes unstable and creates almost fully connected graphs. It is shown later that *secondary* metric is excellent for refinement of the *adaptive* metric but it should not be used on its own.

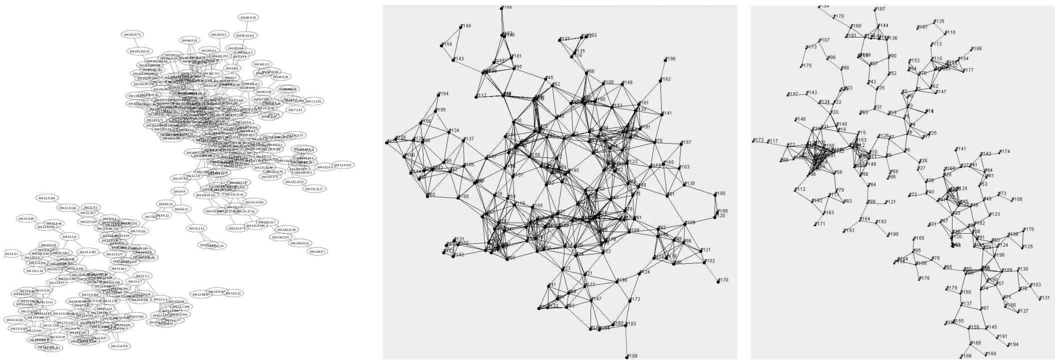
Figures 6 and 7 informally illustrate differences between topologies created by the uniform placement model, sample real topology and topologies created by our algorithm. With increase in average node degree, the uniform placement model creates fewer partitions but in all cases bridges are rare. None of uniform placements resemble the shape of real topology (Figure 7).

If no retries are used (or no metric) our algorithm does not bring improvements: it creates a connected graph with few bridges and articulation points. However, with retries and *combined* metrics ($p = 5, s = 1$) it generates topology that resembles real topologies.

Although visual representation of topologies gives valuable insight in shape of generated topologies, such informal comparison is not sufficient. The following section statistically compares the properties of generated and real topologies.

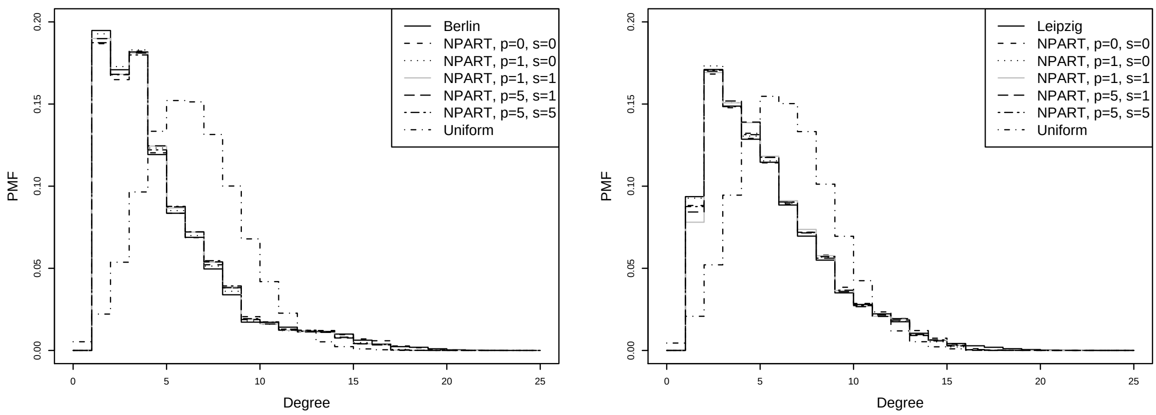
5.1 Properties of generated topologies

Figure 8 shows the vertex degree probability mass function (PMF). As it can be seen, for all parameter combinations, the degree distribution of topologies created by our algorithm precisely



Sample from Berlin's network　　Our algorithm, no retries　　Our algorithm, with retries

Fig. 7. Visual comparison of topologies produced with our algorithm and a real sample. Real topology example is visualized in Graphviz tool [7].



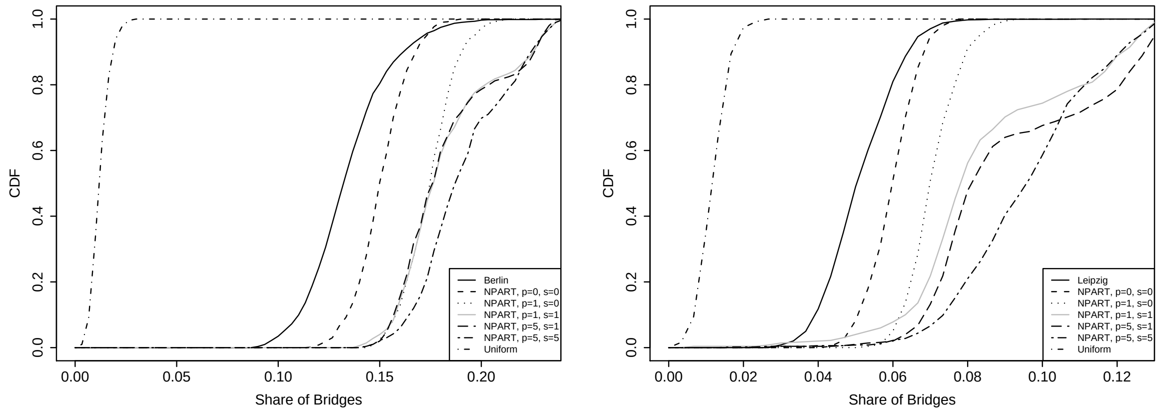
(a) NPART/Berlin　　(b) NPART/Leipzig

Fig. 8. Comparison of node degree distributions.

follows the distribution of real networks. The algorithm adapts with ease to both of distributions. The uniform distribution has its own shape that is considerably different from real distributions. It also has zero-degree nodes, indicating existence of partitioned networks.

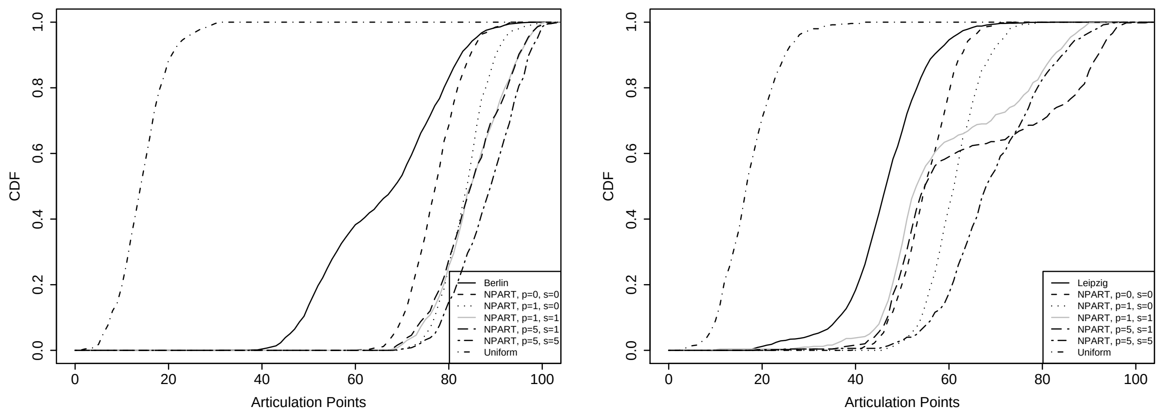
The bridge to edge ratio (Figure 9) and articulation point count (Figure 10) show that NPART topologies follow the properties of real networks. However, the proposed algorithm creates slightly more bridges and articulation points than it should. The uniform placement model is unable to adapt nor to represent the reality: its topologies have less than 1% of bridges and few articulation points.

Figure 11 shows the cumulative distribution of relative component size obtained by removal of bridges. The majority of components are small (less than five nodes) and distribution directly obtained from component size would not provide much information – almost whole distribution weight would be concentrated between one and five. To offset this effect, each component is weighted by its size, relatively to the network size: $C_{rel} = \frac{C_{count} \cdot |C|}{n}$.



(a) NPART/Berlin (b) NPART/Leipzig

Fig. 9. Cumulative distributions of bridge to edge ratio for real samples and created topologies.



(a) NPART/Berlin (b) NPART/Leipzig

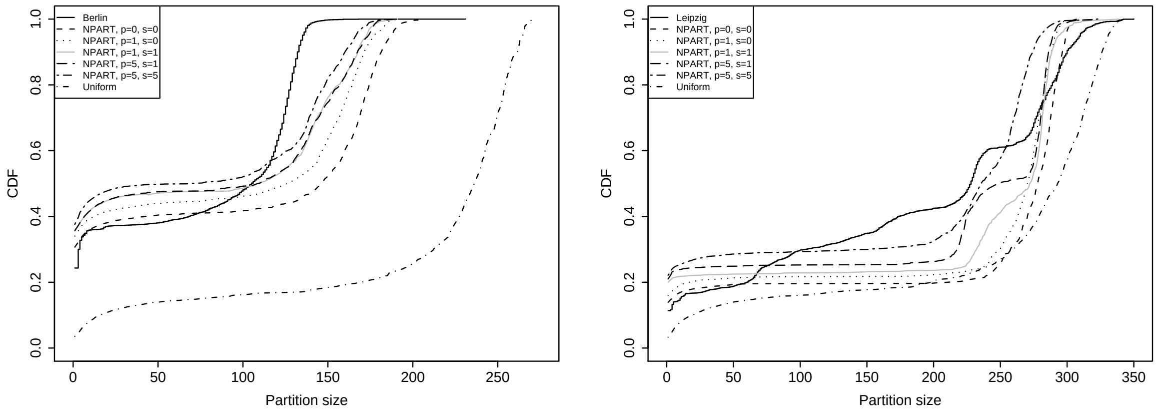
Fig. 10. Cumulative distributions of articulation point count for real samples and created topologies.

For instance, in a graph with 100 nodes, after bridge removal ten components of three vertices exist. The relative component size is: $\frac{3 \cdot 10}{100} = 0.3$: 30% of vertices are in three-vertex components.

NPART is again considerably better than the uniform placement model, in particular for the Berlin's network. Topologies produced with *secondary* metric have distributions more aligned with real measurements than metrics that use only the *adaptive* metric, both for Leipzig and Berlin distributions.

5.2 Offsetting the imprecision brought by simplified modeling

Although considerably better than existing topology generators, the algorithm can be further improved since it creates more bridges and articulation points than it should.



(a) NPART/Berlin (b) NPART/Leipzig

Fig. 11. Cumulative distribution of weighted network components obtained by bridge removal.

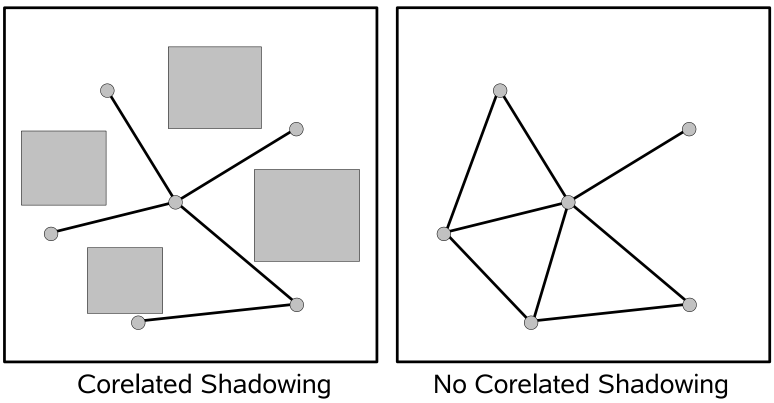
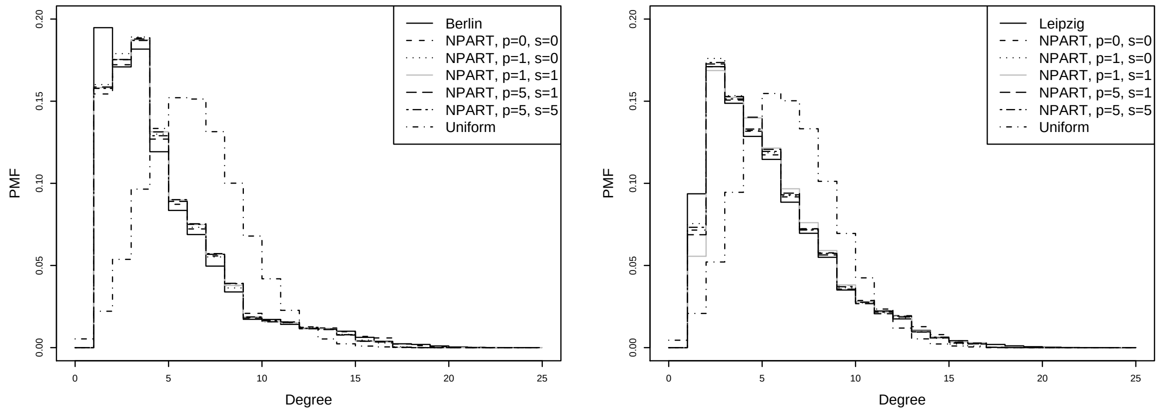


Fig. 12. Impact of correlated shadowing.

We have carefully investigated the input degree data and original topology samples. The analysis revealed a number of nodes with high degree that have lots of pendant nodes attached to it. In reality it is possible due to the phenomenon of correlated shadowing: e.g., a large concrete building blocks all communication links that traverse it, while in its proximity could be open space providing excellent communication possibilities (Figure 12). Correlated shadowing model exists for single wireless hop analysis [8] but is not supported in discrete event simulators for multihop networks nor in our placement algorithm.

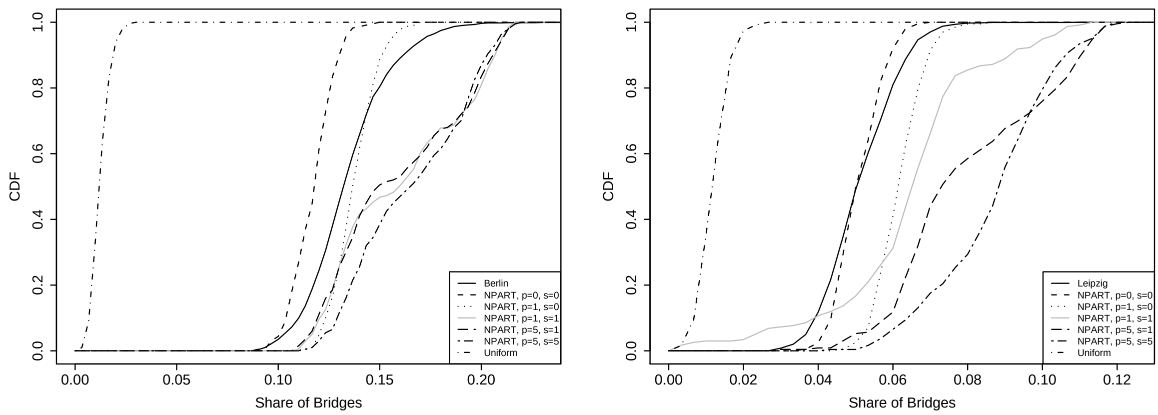
In order to partially offset the lack of correlated shadowing model, we reduce the number of pendant nodes in input distribution by 20%. The reduction of bridge and articulation point count is obvious as shown in Figures 14 and 15 and they are closer to real distributions than in Figures 9 and 10. As expected, the degree distribution (Figure 13) follows closely the real distribution, except of course for nodes of degree 1.

The relative component size distribution in Figure 16 retains good fit with reality as for the original distribution. It also demonstrates the importance of *secondary* metric: in Figure 16(a)



(a) NPART/Berlin (b) NPART/Leipzig

Fig. 13. Comparison of node degree distributions after reduction of pendant node count by 20%.

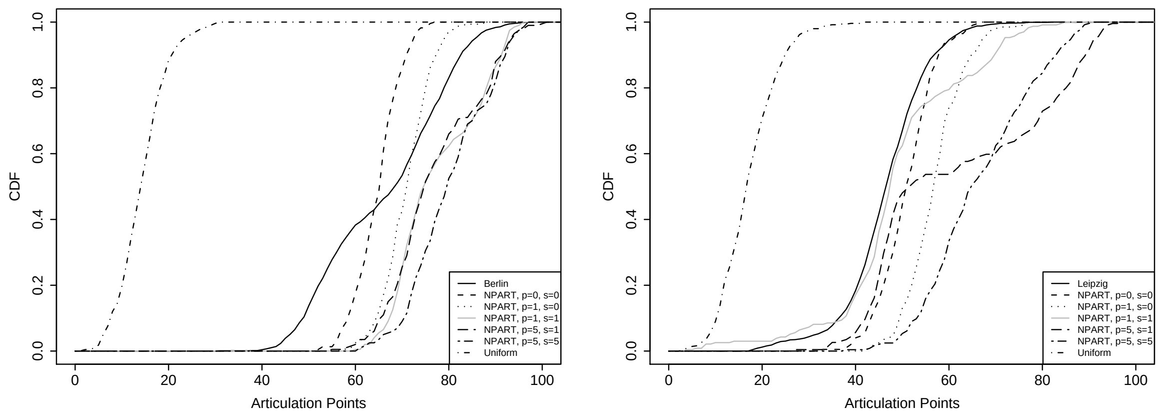


(a) NPART/Berlin (b) NPART/Leipzig

Fig. 14. Cumulative distributions of bridge to edge ratio for real samples and created topologies after reduction of pendant node count by 20%.

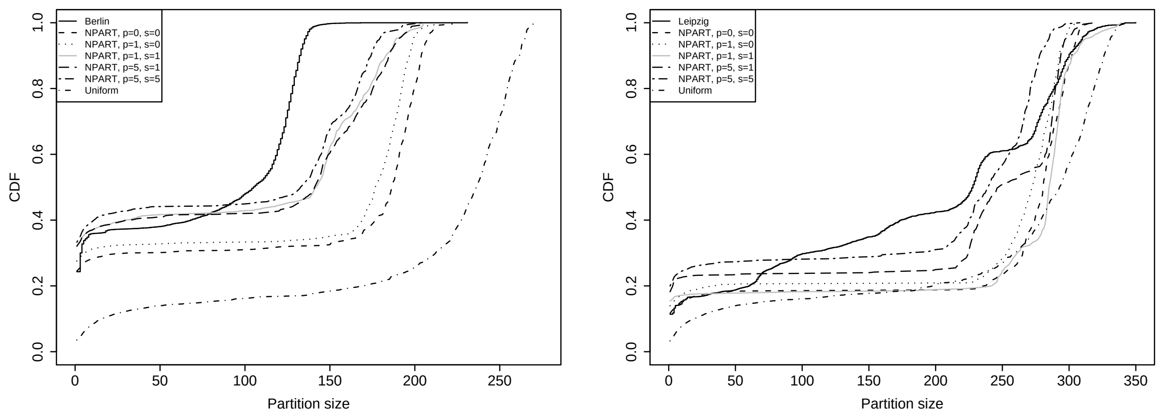
the topologies created without it have poorer alignment with reality than in Figure 11(a), while topologies that have used the secondary metric remain as they were.

In order to summarize and quantify the differences between uniform placement model, NPART algorithm and reality, we have calculated Mean Square Error (MSE) between measurement results and generated topologies. The value of mean square error of a single approach is difficult to interpret: although it is known that a better approach has smaller value of MSE it is not possible to determine a MSE threshold that guarantees acceptable approach. However, MSE is excellent metric for comparison of multiple approaches since it provides their relative ordering when compared to measurements. Tables 3 and 4 show MSE values for distributions of node degree, bridges and articulation points with and without reduction of pendant vertex count. The advantage of



(a) NPART/Berlin (b) NPART/Leipzig

Fig. 15. Cumulative distributions of articulation point count for real samples and created topologies after reduction of pendant node count by 20%.



(a) NPART/Berlin (b) NPART/Leipzig

Fig. 16. Cumulative distribution of weighted network components obtained by bridge removal after reduction of pendant node count by 20%.

NPART is clear as it has by order of magnitude smaller MSE values than the uniform node placement model.

Based on MSE values and Figures 11 and 16 we conclude that parameter combination of $p = 5, s = 1$ provides the best compromise between bridge share and articulation point count (fit with reality decreases with increase in s) and relative component size (fit with reality improves with increase in s).

5.3 Analysis of algorithm's execution time

The complexity of the algorithm is difficult to calculate because it includes stochastic deciding process in it: the innermost loop is repeated until a connected graph is produced. Still, it is

	Degree Berlin	Bridges Berlin	Art.Points Berlin	Degree Leipzig	Bridges Leipzig	Art.Points Leipzig
NPART, p=0, s=0	7.254e-06	0.001295	0.000924	4.042e-06	0.001185	0.001108
NPART, p=1, s=0	2.779e-06	0.002929	0.001768	2.307e-06	0.002959	0.002368
NPART, p=1, s=1	7.370e-06	0.002221	0.001311	2.084e-05	0.002156	0.000625
NPART, p=5, s=1	7.558e-06	0.002298	0.001155	1.233e-05	0.002436	0.000761
NPART, p=5, s=5	9.028e-06	0.002363	0.001650	4.334e-06	0.002334	0.002004
Uniform	3.85e-03	0.007200	0.003224	1.775e-03	0.005367	0.003199

Table 3. Comparison of mean square errors.

	Degree Berlin	Bridges Berlin	Art.Points Berlin	Degree Leipzig	Bridges Leipzig	Art.Points Leipzig
NPART, p=0, s=0	9.187e-05	0.001191	0.002057	2.805e-05	0.000416	0.000461
NPART, p=1, s=0	7.701e-05	0.000580	0.001250	2.281e-05	0.001812	0.001607
NPART, p=1, s=1	8.531e-05	0.000647	0.000865	8.906e-05	0.001258	0.000251
NPART, p=5, s=1	8.422e-05	0.000547	0.000720	4.342e-05	0.001622	0.000629
NPART, p=5, s=5	8.491e-05	0.000828	0.000811	2.580e-05	0.002210	0.001641
Uniform	3.85e-03	0.007200	0.003224	1.775e-03	0.005367	0.003199

Table 4. Comparison of mean square errors after pendant node count reduction.

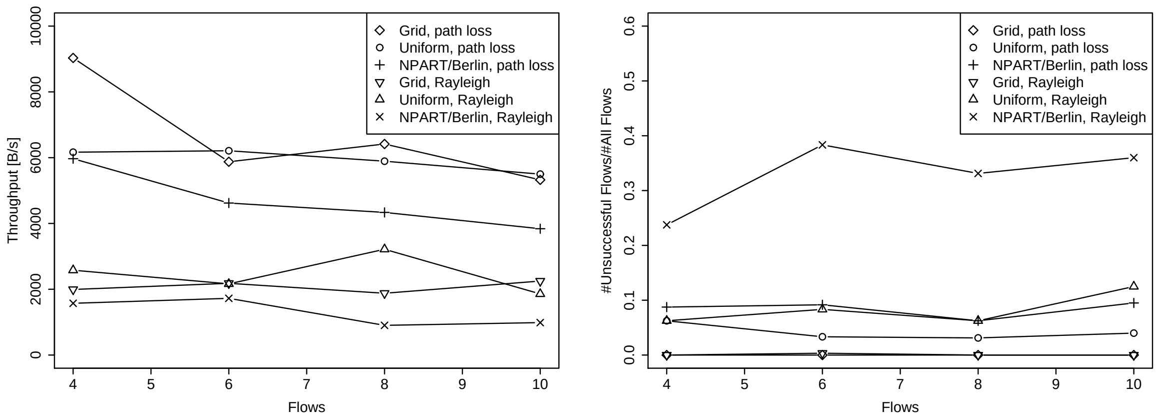
necessary to evaluate the time algorithm needs to create a topology in order to demonstrate the algorithm behavior that can be experienced by a user.

For evaluation we measure the time needed to generate a topology. The test server has 32GB memory and 8 Dual-core AMD Opteron processors working at 2.6GHz. However, the algorithm implementation uses only one processor core at a time. The algorithm is implemented in Java programming language and run in Sun's Java 2 Runtime Environment Standard Edition (build 1.5.0_06-b05). The measurements in this section are based on 500 executions of the algorithm.

As a reference point for time execution of the algorithm, we have also implemented the uniform placement algorithm and run it on same test computer under same conditions. The task was to produce a connected graph consisting of 275 vertices with average vertex degree of 4.17 (numbers are equal to the average degree and average number of participating nodes in main partition of Berlin's network). The time required to place nodes and to create connectivity graph is measured. Connectivity testing is not included in measured time. At first, we tried to evaluate time required for fully connected graph (all 275 vertices belong to the same graph component). However, after several hours of attempts, the uniform placement algorithm did not produce even a single fully connected topology. The connectivity condition was then weakened, and we have measured time required for creation of topologies whose biggest partition contains at least 97.5% of placed nodes.

With the weaker condition the uniform placement algorithm needs 489.8 seconds per topology on the average. Although the uniform placement algorithm is substantially faster for a single execution, for low-degree networks it requires more time. The topologies that it creates are partitioned, and on the average 66623 retries are required before the 97.5% connectivity condition is met.

NPART is considerably faster for such sparse graphs. If only *adaptive* metric is used, topology is created in only 2.84 seconds on the average. The *combined* metric is more demanding and if it is used, the average execution time grows to 288.56 seconds. Still, that is 41.1% faster than



(a) Throughput.

(b) Ratio of unsuccessful flows to total number of flows.

Fig. 17. ns-2 simulation results for different topologies and signal propagation models. AODV routing protocol is used.

the uniform algorithm with the 97.5% connectivity constraint. For sparser graphs, it is to expect that difference increases considerably in favor of NPART.

5.4 ns-2 Simulation results

This section demonstrates that choice of node placement algorithm considerably impacts simulation results. For this purpose, we have used ns2 simulator [5], version 2.29 with Rayleigh-Ricean fading extension [16]. Nominal communication range of nodes is set to 250m.

There are six distinct simulation setups:

- Grid node placement and path-loss propagation model
- Grid node placement and Rayleigh propagation model
- Uniform node placement and path-loss propagation model
- Uniform node placement and Rayleigh propagation model
- NPART/Berlin node placement and path-loss propagation model
- NPART/Berlin node placement and Rayleigh propagation model

Grid consists of 272 nodes, put in 16 rows and 17 columns. Internode distance is 200m. For uniform node placement, 275 nodes are placed in 2700x2700m area, producing average node degree of 7.4. Such parameter selection creates network that is not too dense but connected with rather high probability. NPART algorithm generates 275-node topologies, using data input from Berlin's network and combined metric (s=1, p=5). Routing protocol is AODV [15]. Its default parameters have not been changed.

In each of simulation setups the number of TCP flows is varied (4,6,8,10). Throughput and number of flows that were unable to start (unsuccessful flows) are measured. In order to avoid counting of unsuccessful flows that are caused by partitioned network, uniform placement topologies are tested for connectivity before they are accepted for simulation.

Each point in Figure 17 is calculated from 50 simulations performed on 50 different topologies (except for grid where all topologies are identical). TCP flows are created between randomly selected pairs of nodes. Warm up phase is set to 30s and simulation is executed 250s.

As expected and already shown in research studies, there exists a considerable difference in simulation results between over-simplistic, over-optimistic path-loss propagation model and more realistic Rayleigh model: obtained throughput is considerably higher and number of unsuccessful flows is considerably lower for the path-loss model.

It is important to observe that for the same propagation model there also exists considerable difference in simulation results between grid, uniform node placement and NPART produced topologies: in NPART topologies throughput is lower than in both synthetic placements.

Simulation of NPART/Berlin topologies with realistic wireless signal propagation model results in particularly high ratio of unsuccessful flows – although a communication path between pair of nodes exists (NPART guarantees connected topologies), AODV is unable to find it. In [11] we have predicted such behavior of AODV with different methodology.

Uniform placement model creates fewer unsuccessful flows than NPART/Berlin while in grid structured network only one of all simulated flows was unsuccessful, even with Rayleigh fading on the wireless channel.

6 Conclusions

We have proposed, developed and evaluated NPART - Node Placement Algorithm for Realistic Topologies. The algorithm provides input for simulation of static wireless multihop networks. It is flexible since it is sufficient that user defines a metric in accordance with his/her needs or provides different data input set and algorithm creates topologies with different properties. The algorithm guarantees the connectivity of produced topologies.

We have evaluated four metrics with the algorithm and shown that with appropriate metric and parameter selection NPART creates realistic topologies. Stochastic analysis is used to compare NPART-produced topologies with ubiquitous uniform placement algorithm and real networks in Berlin and Leipzig. The properties of interest are node degree distribution, bridge to edge ratio, articulation point count and size of graph components after bridge removal. The properties of topologies produced by our algorithm fit closely with measurements, while uniform placement model has its own properties that are far from reality.

The importance of accurate signal propagation models is known and shown in numerous publications but the impact of topology generators on simulation outcome is often overlooked. In order to demonstrate importance of node placement models in simulation, we have compared throughput and number of unsuccessful flows in topologies produced by the grid, uniform placement model and our proposed NPART algorithm under simplistic path-loss and realistic Rayleigh signal propagation models. Simulation results show that node placement model plays as important role in simulation outcome as the wireless signal propagation model.

We hope that our results will encourage the research community to use realistic modeling in all segments of simulation setup thus increasing the simulation quality and narrowing the gap between simulation and reality.

References

1. A. Aguiar and J. Gross. Wireless channel models. *Technical Report TKN-03-007*, 2003.
2. C. Bettstetter. On the minimum node degree and connectivity of a wireless multihop network. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking and computing*, Lausanne, Switzerland, 2002.
3. C. Bettstetter, M. Gyarmati, and U. Schilcher. An inhomogeneous spatial node distribution and its stochastic properties. In *Proceedings of 10th ACM-IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM 2007)*, Chania, Greece, 2007.
4. K. L. Calvert, M. B. Doar, and E. W. Zegura. Modeling internet topology. *IEEE Communications Magazine*, 35(6):160–163, June 1997.
5. K. Fall and K. Varadhan. The ns2 manual. 2008. http://www.isi.edu/nsnam/ns/ns-documentation.html.

6. L. Fang, W. Du, and P. Ning. A beacon-less location discovery scheme for wireless sensor networks. In *In Proceedings of 24th Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM2005*, 2005.
7. E. Gansner and S. North. An open graph visualization system and its applications to software engineering. *Software Practice and Experience*, 30(11):1203–1233, 2000.
8. M. Gudmundson. Correlation model for shadow fading in mobile radio systems. *Electronic Letters*, 27:2145–2146, 1991.
9. X. Li, P. Wan, Y. Wang, and C. Yi. Fault tolerant deployment and topology control in wireless networks. In *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking and computing*, Maryland, USA, 2003.
10. X. Liu and M. Haenggi. Toward quasiregular sensor networks: Topology control algorithms for improved energy efficiency. *IEEE Transactions on Parallel and Distributed Systems*, 17:975–986, 2006.
11. B. Milic and M. Malek. Adaptation of breadth first search algorithm for cut-edge detection in wireless multihop networks. In *Proceedings of 10th ACM-IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM 2007)*, Chania, Greece, 2007.
12. B. Milic and M. Malek. Analyzing large scale real-world wireless multihop network. *IEEE Communication Letters*, 11:580–582, 2007.
13. B. Milic and M. Malek. *to appear in : Handbook of Wirless Ad Hoc and Sensor Networks*, chapter Properties of wireless multihop networks in theory and practice. Springer Verlag, 2008.
14. F. A. Onat and I. Stojmenovic. Generating random graphs for wireless actuator networks. In *Proceedings of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2007.*, 2007.
15. C. Perkins, E. Belding-Royer, and S. Das. Ad hoc on-demand distance vector (AODV) routing (RFC 3561). July 2003.
16. R. J. Punnoose, P. V. Nikitin, and D. D. Stancil. Efficient simulation of ricean fading within a packet simulator. In *Proceedings of the Vehicular Technology Conference*, 2000.
17. N. Vicari. Models of WWW-Traffic: a Comparison of Pareto and Logarithmic Histogram Models. Technical Report 198, Institute of Computer Science, University of Wuerzburg, 1998.
18. D. B. West. *Introduction to Graph Theory.* Prentice Hall, 1996.
19. E. W. Zegura, K. L. Calvert, and M. J. Donahoo. A quantitative comparison of graph-based models for Internet topology. *IEEE/ACM Transactions on Networking*, 5(6):770–783, 1997.