On Availability-Performability Tradeoff in Wireless Mesh Networks

Parth H. Pathak, Rudra Dutta, and Prasant Mohapatra

Abstract—It is understood from past decade of research that a wireless multi-hop network can achieve maximum network throughput only when its nodes operate at a minimum common transmission power level that ensures network connectivity (availability). This point of optimality where maximum availability and throughput is guaranteed in an interference-optimal network has been the basis of numerous design problems in wireless networks. In this paper, we claim that when performability (availability weighted performance) is considered as opposed to average case throughput performance, there does not exist a transmission power (or node density) that can maximize both availability and performability. Since the current mesh networks are expected to deliver carrier-grade services to its users, the availability-performability tradeoff presented in this paper holds a special importance. While availability metric is a necessary one for any networking system intended to provide continuous service, past research [1] has shown a strong correlation between performability and quality of user experience in case of wireless networks. The contributions of the paper are as follows: (1) We first define availability and performability in the context of wireless mesh networks, and then develop efficient algorithms on the basis of intelligent state sampling that can calculate both the quantities with reasonable accuracy. (2) We apply the evaluation methods to two existing mesh networks (GoogleWiFi [2] and PoncaCityMesh [3]) to demonstrate that their current design can not guarantee a reasonable level of availability or performability. (3) Using hundreds of hours of simulations, we analyze the impact of two basic deployment factors (node density and transmission power) on availability and performability. We outline numerous novel results that emerge due to joint availability-performability analysis including the observation about availability-performability tradeoff.

Index Terms—Wireless mesh networks, service continuity, availability, performability, deployment factors

1 Introduction

It is generally acknowledged due to many previous seminal research works (such as [4], [5]) that a wireless multi-hop network achieves optimal throughput performance only when its nodes operate at a common transmission power that is the minimum required to retain network connectivity. This is shown in an example in Fig. 1. Network design that can realize this sweet spot of maximum availability and performance has been the basis of numerous research efforts in wireless mesh networks (WMN) over the last decade.

In this paper, we claim that when performability (availability weighted performance) is considered as opposed to the absolute performance, availability and performability are at odds with each other. This means that a WMN's performability becomes sub-optimal when trying to maximize the availability (and vice versa). Here, availability is the fraction of the time a WMN is available to its users, while performability is a composite measure of performance (here—network throughput) and robustness. As opposed to average-case performance which is commonly studied in literature, performability captures the value of performance that can be guaranteed even in presence of failures. It has been shown in [1] that performability of a performance degradable system such as WMN strongly correlates to aggregate experience of users.

The performability and availability analysis holds a special importance for WMNs because current mesh deployments are expected to deliver carrier-grade services to its users. Current WMNs are being designed to handle emergency services [3], surveillance [6], smart-grid operations [7] etc. It is absolutely essential that a mesh network continues to operate with high availability and performability even in presence of random link or node failures. With these increasing expectations, it is generally being recognized that service continuity metrics such as availability and performability are not well-understood for WMNs. This is natural since ad-hoc networks (from which the mesh concept was derived) were never designed to provide carrier-grade services. WMNs, on the other hand, must be analyzed in the same context as other contenders such as cellular networks. For example, there already exists various methods for evaluating service continuity for cellular networks (e.g., [8], [9]) in case of component failures.

In this paper, we first define availability and performability in the context of WMNs. While their definitions are more or less intuitive, we show that evaluating them precisely is extremely complex. We then present efficient algorithms that can estimate the quantities with bounded confidence. Both, availability and performability, are then studied for two basic topology deployment factors—transmission power and node density. While varying topology factors (tx-power or node density), throughput performance and availability attain their maximum at a common tx-power (or node density). Although this is true for

- *P.H. Pathak and P. Mohapatra are with the Computer Science Department, University of California, Davis, CA 95616. E-mail: {phpathak, pmohapatra}@ucdavis.edu.*
- *R. Dutta is with the Computer Science Department, North Carolina State University, Raleigh, NC 27695. E-mail: dutta@csc.ncsu.edu.*

Manuscript received 13 Dec. 2013; revised 15 May 2014; accepted 20 May 2014. Date of publication 11 June 2014; date of current version 29 Jan. 2015. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TMC.2014.2329845

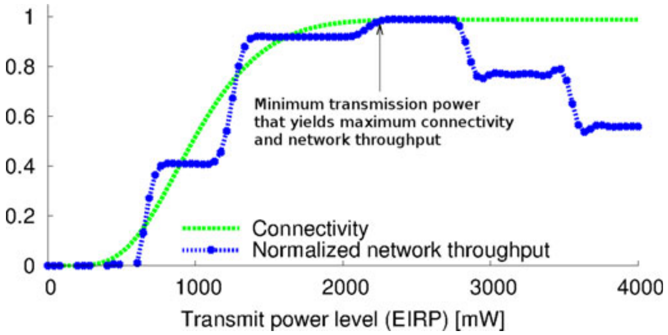


Fig. 1. An example mesh network with 100 mesh nodes and 10 gateways (uniformly distributed); uniform node-to-gateway traffic over shortest path to nearest gateway; connectivity defined as the fraction of nodes that can reach a gateway; no link failures assumed; 802.11a MAC with fixed link rate in NS-3.

throughput performance, our study establishes that throughput performability and availability attain their maximum at different tx-power and node density values. This is because topology factors necessary for maximum availability already have a detrimental effects of network throughput performance even when rest of the network settings are interference-optimal. This shows that maximum performability is achieved when collision domains of gateways interfere with each other just enough so that availability advantage of the overlap is more as compared to throughput penalties.

We believe that this evaluation and analysis of availability-performability trade-off is crucial in understanding how a realistic service-level agreement (SLA) can be established for current-age WMNs. We target tx-power and node deployment in our study because their impact on topology formation and network capacity is the most fundamental in nature. It has been shown that other recent ways of performance improvement (such as channel assignment, MIMO etc.) remain orthogonal to their impact.

Availability and performability are described as follows in the context of WMNs.

(1) K-center availability (KCA). The availability metric measures what fraction of time a given system is in working state. We name the availability of WMNs to be k-center availability as the Internet gateways of WMNs are ideally the centers of mesh network connectivity graph. *The KCA problem finds the probability that every mesh node is connected to at least one of the k Internet gateways.* KCA is an essential service continuity metric since it models if WMN as a system is available to its users or not. Current mesh network users expect an "always-on" attribute in service where total disruption is seen as a serious failure of the network. Maximizing the fraction of the time a mesh can provide Internet access to its users should be considered as an imperative design objective.

(2) K-center performability (KCP). The performability metric is a joint measure of system performance and its availability. A WMN that enables continuous Internet connection of its users but does so with frequent lapses in performance (such as throughput, delay etc.) can not provide a satisfactory user experience. *The KCP metric accounts for what is the performance of a given WMN and how reliably the performance is delivered to its users.* Increasing the KCP of WMNs ensures that users encounter fewer and fewer variations in the performance that they expect from WMN.

The contributions of the work are as follows:

- First, we formally define KCA and KCP. It is shown that their exact evaluation for even a moderate size WMN incurs prohibitive computational complexity. We develop efficient algorithms using informed state sampling that can estimate the quantities with reasonable accuracy even for a large urban-scale WMN (as many as 500 nodes).
- Second, we apply these methods of evaluating KCA and KCP to two existing urban mesh networks (GoogleWiFi [2] and PoncaCityMesh [3]). We show that their current design can not provide a reasonable level of availability and performability.
- Third, in order to further understand the impact of deployment factors on KCA and KCP, we embark on a comprehensive study of their relationship using hundreds of hours of simulations. *Using the results, we identify a novel fact that the values of tx-power (or node density) that can maximize availability and performability are different. This is in sharp contrast to design principles that are followed conventionally which suggest that operating a WMN at a tx-power that is minimum required to ensure availability can yield maximum average-case performance. In fact, we show that it is advisable to operate the network with overlapping collision domains of gateways because the availability benefits of doing this is more compared to performance penalties associated with it.*

The paper is organized as follows:

- *Section 2*. KCA definition and evaluation algorithm,
- *Section 3*. KCP definition and evaluation algorithm,
- *Section 4*. KCA analysis of GoogleWiFi, PoncaCityMesh and interference-optimal mesh networks,
- *Section 5*. KCP analysis of GoogleWiFi, PoncaCityMesh and interference-optimal mesh networks,
- *Section 6*. KCA-KCP tradeoff,
- *Section 7*. Accuracy and efficiency analysis of KCA and KCP evaluation methods,
- *Section 8*. Conclusions.

2 KCA Evaluation

2.1 Network Model

Formally, let $G_P(V, E, P)$ be the network graph of a mesh network. let n be the number of mesh nodes and m be the number of edges. For the edge set $E = \{e_1, e_2, \ldots, e_m\}$, let $P = \{p_{e_1}, p_{e_2}, \ldots, p_{e_m}\}$ be their availabilities (probability of correct operation). A link is available/operational when it can transfer data at its default pre-configured data rate, and is assumed to be unavailable/failed if it can not transfer any data. We do not consider the case where links operate at lower data rates due to mutual interference but instead tackle the problem using an interference model that ensures that only non-interfering set of links are active at any point of time. This ensures that any link is either in operational or failed mode, and restricts the total state space to 2^m states.

Here, we consider that the links fail due to shadow fading. This is a common form of link failure especially in urban environment [10] which results into non-uniform propagation of signals. With shadow fading, availability of a link uv can be given by

$$\Pr\{P_v^u \geq P_{min}\} = \frac{1}{2}\left[1 - \mathrm{erf}\left(\frac{10}{\sqrt{2}} \cdot \log_{10}\frac{d_{uv}}{d_r} \cdot \frac{\eta}{\sigma}\right)\right], \quad (1)$$

where P_v^u is the received signal power from u to v, P_{min} is the receiver sensitivity, d_{uv} is the distance between nodes u and v, d_r is the communication range of nodes in absence of shadowing, η is the path-loss exponent, and σ is the standard deviation of zero-mean Gaussian variables representing shadowing. We use Eq. (1) to calculate the link availabilities of a given RF profile.

Note that the methods described next for evaluating KCA and KCP *apply to any probabilistic connectivity graph G_P irrespective of how the link availabilities are obtained*. Typically, availability of links can be obtained using empirical data or can be derived using well-known propagation models. In our work, we consider the shadow fading model for obtaining link availability values because shadow fading is identified as a crucial design factor in outdoor wireless mesh networks [10], [11], [12]. Evaluating our methods of determining KCA and KCP for more complex link models (such as time-varying channels and multipath fading) is very important and we leave this to our future work. For this paper, we only focus on link availability based on shadow fading as described above.

Next, we define KCA and provide the details of constrained Monte Carlo simulation (CMCS) method for estimating its value.

2.2 K-Center Connected Network State

For G_P, let $K \subset V$ be a set of k gateways where $K = \{g_1, g_2, \ldots, g_k\}$. K-center Availability is the probability that $\forall u \in V - K$ is connected to at least one gateway $g \in K$. When all non-gateway mesh nodes can reach at least one gateway, we mark the network to be *k-center connected*.

Different from various other approaches in literature that focus on deriving asymptotic results [13], [14], [15] for network connectivity, *we are interested in numerical evaluation of KCA*. Also, network availability can not be used for estimating KCA in any way because there can be a large number of network states where network is disconnected due to a link failure but the state is still k-center connected.

2.3 Constrained Monte Carlo Simulation

Since there are a total of 2^m network states of a WMN, their complete enumeration is extremely expensive for even a small value of m. Traditionally, either bounds or Monte Carlo sampling is used for estimating availability. Bounds can be derived by exploiting some structural properties of graph, but they are often times too loose to be useful. Monte Carlo state generation uniformly chooses states from the complete state space, and estimates the availability using them. Here, the advantage of accuracy comes at the cost of efficiency. Fishman first proposed an availability evaluation scheme [16] (for two-terminal availability) that combined benefits of both these approaches. He suggested that even if the bounds found are only reasonably good, they can be used to direct a Monte Carlo simulation to reduce the number of necessary samples. This means that bounds should be used to eliminate large number of states from the entire state space, and Monte Carlo simulation is then necessary to be ran only in remaining number of states. This is shown in Fig. 2. With the bounds, we already know that all the states above the upper bound are connected and all the states below the lower bound are disconnected. Now, if we only generate the states in between the bounds (undetermined states) in our Monte Carlo simulation, fewer states will be necessary as compared to a naive Monte Carlo simulation. We refer to this scheme as a *Constrained Monte Carlo Simulation.*

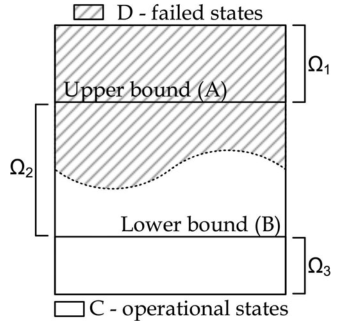


Fig. 2. State space and availability bounds.

2.4 KCA Evaluation Using CMCS

In this section, we show how CMCS can be used for KCA estimation. First, we show how edge-packing upper and lower bounds on KCA are calculated, and then describe the complete CMCS procedure.

Let $S = \{s_{e_1}, s_{e_2}, \ldots, s_{e_m}\}$ describe a network state where $s_{e_i} = 1$ if edge e_i is operational (available) and 0 otherwise. The state space with total of 2^m states is divided into two disjoint and exhaustive subsets $\mathcal{C}$ and $\mathcal{D}$. As shown in Fig. 2, $\mathcal{C}$ is a set of all operational (k-center connected) states while $\mathcal{D}$ is a set of all failed (k-center disconnected) sets. We use edge-packing bounds which are described next.

2.4.1 Edge-Packing Bounds

In a graph $G = (V, E)$, an *edge packing* [17] of G by k graphs $G_1, G_2, \ldots, G_K$ is obtained by partitioning the edge set E into $k+1$ classes $E_1, E_2, \ldots, E_k, U$ and defining $G_i = (V, E_i)$. This way, an edge-packing of a graph is a collection of edge-disjoint subgraphs of the graph.

An edge-packing lower bound on the availability can be obtained by finding a set of *edge-disjoint minimal paths* (edge-packing of minpaths), and obtaining the probability that all the edges of at least one of these paths operate correctly.

An edge-packing upper bound on the availability can be obtained by finding a set of *edge-disjoint minimal cuts* (edge-packing of cuts) and finding the probability that all the edges of at least one of these cuts fail.

The edge-disjointness ensures that failure of every path (or occurrence of any cut) is independent from others since failure of an edge will only impact one path (or cut) at most.

Lower bound. For KCA, edge-packing of minimal pathsets is a set of edge-disjoint forests where in each forest, every tree contains a gateway and all mesh nodes belong to a tree. Let I be the total number of such forests which are denoted by $F_1, F_2, \ldots F_I$. Each forest is a union of a minimal path

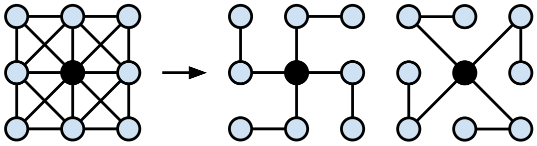


Fig. 3. When one of the two trees is operational, the network is in k-center connected state.

from every mesh node to a gateway. If all edges of at least one of these forests are operational, every mesh node can reach at least one gateway and the resultant state is operational. Let Ω_1 be the set of states where there exists at least one operational forest. By definition, Ω_1 is a subset of $\mathcal{C}$ as shown in Fig. 2. Hence, a lower bound (B) on k-center availability is the probability that all the edges of at least one of these forests operate correctly

$$B = 1 - \prod_{1 \le j \le I} \left(1 - \prod_{e_i \in F_j} p_i \right). \tag{2}$$

Finding I edge disjoint forests where trees in each forest are rooted at gateways turns out to be a non-trivial task. This is because as per the Eq. (2), more number of such edge disjoint forests (maximize I) yields a better bound. Also, the lower bound improves if each forest contains edges which are more reliable (higher operational probability) compared to other non-forest edges.

To find this edge packing of forests, we first determine weight of every edge using $w_{e_i} = -\ln(p_{e_i})$. Now, let $d(u,v)$ be the total weight of all edges on the shortest path between u and v. We then partition G_P in subgraphs $G_{g_1}, G_{g_2}, \ldots, G_{g_k}$ where $v \in G_{g_i}$ if $d(v, g_i) = \min\{d(v, g_i), g_i \in K\}$, and $e_i \in G_{g_i}$ if and only if both endpoints of e_i belong to the same subgraph. Let $\kappa(G_{g_i})$ be the edge connectivity of G_{g_i} and for graph G_P let $\kappa(G_P) = \min\{\kappa(G_{g_i}), g_i \in K\}$. Now from Tutte's theorem [18], $I = \lfloor \kappa(G_P)/2 \rfloor$ and G_P contains at least I edge disjoint forests where each forest contains a tree belonging every subgraph G_{g_i}. Once we have determined I, we can find these edge disjoint forests using method proposed in [19]. To find I forests in which every node can reach a gateway using fewer and more reliable edges, we first find cw_{e_i} score of every edge in a G_{g_i}. That is, $\forall e_i \in G_{g_i}$ between u and v, let $cw_{e_i} = \min\{d(u, g_i), d(v, g_i)\} + w_{e_i}$. Now, we sort all $e_i \in G_{g_i}$ in increasing order of their cw_{e_i} scores, and input the sorted list to matroid partition algorithm of [19] which then finds I edge disjoint trees in each G_{g_i} forming I edge disjoint forests.

Fig. 3 shows a simple example of a single-gateway mesh network, and how it can be divided into two edge-disjoint trees.

Upper bound. There can be many different ways of determining a set of minimal cuts which can cause a failed network state. We observe that all adjacent edges of every node in an independent set of the graph forms a minimal cut, and all such cuts are also edge-disjoint. Since any such cut disconnects a node from rest of the network, such a node can not be connected with any of the gateway which is a failed state as per the definition. Let J be the total number of such cuts which are denoted by $C_1, C_2, \ldots, C_J$. If all the edges of at least one of these cuts are failed, the resultant network state is failed. Let $\Omega_3 \subset \mathcal{D}$ be the set of states where there exists at least one failed cut. The upper bound (A) on k-center availability is the probability that all edges of at least one of these cuts have failed

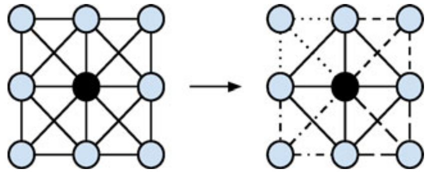


Fig. 4. When all edges of a cut fail, the network is in k-center disconnected state.

$$A = \prod_{1 \le j \le J} \left(1 - \prod_{e_i \in C_j} (1 - p_i) \right). \tag{3}$$

We use well-known greedy algorithm for finding a maximal independent set, and then obtain the cuts as above. Fig. 4 shows how we can find four edge-disjoint cuts using the method described above.

Sampling algorithm. As in Fig. 2, since we already know that all states in Ω_1 are connected and all states in Ω_3 are disconnected, only the states in between the upper bound and lower bound are undetermined, and Monte Carlo simulations are required to determine their status. Let Ω_2 be the set of all such states. Note that Ω_2 is the set of states such that every cut C_j where $1 \le j \le J$ is operational and every forest F_i where $1 \le i \le I$ has failed. Since all cuts C_j are operational, there can be an operational path from every mesh node to a gateway but union of all such paths is not a forest among F_i.

3 KCA Evaluation Algorithm

We now describe the sampling algorithm for KCA estimation which is adapted from [16], [20]. The complete detailed algorithm is provided in Algorithm (1). In the sampling algorithm, the states of Ω_2 can not be uniformly sampled as in the naive approach because status of edges in a cut of a forest is conditionally dependent on each other. There are two steps of sampling when sampling a state in Ω_2

1) The edges which belong to a forest and/or a cut have to be sampled sequentially. If an edge belongs to a forest, it status depends on the constraint that requires at least one edge of the forest to fail. This ensures that the forest is not operational. Similarly, if an edge belongs to a cut, its status depends on the constraint that requires at least one edge of the cut to be operational. This ensures that the cut has not failed.
2) The edges not belonging to a forest or a cut can be sampled independently.

The number of states required to be sampled from Ω_2 depends on the level of accuracy needed in KCA estimate. We use the standard deviation of KCA estimate as a guideline. Since the standard deviation decreases as more and more samples are generated, its value can be calculated after every fixed number of samples. If the targeted accuracy is achieved, no more samples are necessary and procedure can be terminated with confidence.

Algorithm 1: CMCS procedure to calculate KCA

Purpose: To calculate k-center reliability (KCA)
Input:
Graph $G_P(V, E, P)$, $n = |V|, m = |E|$,
Set $K \subset V$ of k gateways, $K = \{g_1, g_2, \ldots, g_k\}$,
$P = \{p_{e_i}, \forall e_i \in E\}$ probabilities of correct operation;
Edge set of I forests $F_1, F_2, \ldots F_I$ and
Lower bound $B = 1 - \prod_{1\le j\le I}(1 - \prod_{e_i\in F_j} p_{e_i})$,
$H_1 = \bigcup_{j=1}^{I} F_j$;
Edge set of J cuts $C_1, C_2, \ldots C_J$ and
Upper bound $A = \prod_{1\le j\le J}(1 - \prod_{e_i\in C_j}(1 - p_{e_i}))$,
$H_2 = \bigcup_{j=1}^{J} C_j$, $H = H_1 \cup H_2$;
For $1 \le i \le m$ and $1 \le k \le I$,
$Forest[e_i] = k$ if $e_i \in F_k$ and 0 otherwise,
$\lambda[0] = 0$ and $\lambda[k] = \prod_{e_i\in F_k} p_{e_i}$;
For $1 \le i \le m$ and $1 \le k \le J$,
$Cut[e_i] = k$ if $e_i \in C_k$ and 0 otherwise,
$\omega[0] = 0$ and $\omega[k] = \prod_{e_i\in C_k}(1 - p_{e_i})$;
Accuracy threshold α;
Output: k-center reliability (KCA)
Method:
$S = 0$;
while *stddev(KCA)* $\le \alpha$ **do**
$X = X + 1$;
Generate a sample state using SAMPLESTATE;
Check if the sample state is k-center connected;
If *k-center connected* **then**
$S = S + 1$;
end
$stddev(KCA) = \sqrt{\frac{(A-B)^2(1-S/X))(S/X)}{(X-1)}}$
end
Compute the k-center reliability as follows
$KCA = B + (A - B)\frac{S}{X}$
return KCA.

PROCEDURE SAMPLESTATE:
// Sample edges in H - Part 1
while $\lambda[k] \ne 0$ *for* $1 \le k \le I$ *and*
$\omega[j] \ne 0$ *for* $1 \le j \le J$ **do**
Select an edge e_i from a F_k or a C_j with fewest remaining edges and $\lambda[k] \ne 0, \omega[j] \ne 0$;
$a = Forest[e_i], b = Cut[e_i]$;
$p^*_{e_i} = \frac{p_{e_i} - \lambda[a]}{1-\omega[b]-\lambda[a]}$;
Sample u from uniform distribution $U[0, 1]$;
$s_{e_i} = \lfloor u + p^*_{e_i} \rfloor$;
set $\lambda[a] = \frac{s_{e_i}\cdot\lambda[a]}{p_{e_i}}, \omega[b] = \frac{(1-s_{e_i})\cdot\omega[b]}{1-p_{e_i}}$;
end
// Sample edges in $E - H$ - Part 2
forall the $e_i \in E - H$ **do**
Sample u from $U[0, 1]$;
$s_{e_i} = \lfloor u + p^*_{e_i} \rfloor$;
end
return the generated network state $(s_{e_1}, \ldots, s_{e_m})$;

4 KCP EVALUATION

We now discuss why KCP evaluation is necessary for WMNs, and describe an efficient method for estimating KCP of large urban-scale mesh networks.

4.1 Why KCP?

For any given state of the network, KCA yields a 0/1 evaluation of whether all the mesh nodes are connected to a gateway or not. Though this evaluation is crucial in providing highly "available" services to users, it gives little or no information about the actual network performance. As an example, even in a k-center disconnected state, a few users might be able to successfully connect to Internet. On the other hand, a k-center connected state ensures connectivity of mesh users to Internet but does not guarantee any specific performance. This requires treating mesh networks as *performance degradable system* instead of denoting it to be operational or failed. Peformability metric is typically used for such a purpose in network survivability studies. As opposed to absolute values of average-case performance, performability captures not only what is the performance but also how reliably it is delivered.

Let $\mathcal{X}$ be the set of all 2^m network states. Also, let $F(S)$ be a performance function defined on the state S. The probability that the system is in state S is given by $\Pr(S)$ (also called state occurrence probability). The well-known performability metric can be calculated as

$$\overline{P} = \Sigma_{S\in\mathcal{X}}(F(S)\cdot \Pr(S)). \quad (4)$$

Note that in cases like KCA problem, $F(S)$ can be binary (S is k-center connected or not) but $F(S)$ can take any form in general which makes the performability evaluation problem even more challenging. Its exact evaluation is known to be a hard problem [21] and most of the current evaluation schemes depend on state generation methods.

4.2 KCP for WMNs

For calculating performability of mesh networks, an obvious choice of $F(\cdot)$ is its aggregate network throughput. For given traffic demand and network configuration (such as routing scheme, MAC protocol etc.), mesh network yields a particular value of network throughput. Since the traffic always flows between mesh nodes and gateways in a mesh network, we refer to its performability as k-center performability.

KCP evaluation is even more difficult because it requires an additional step of throughput assessment of each generated state which is known to be a hard problem for WMNs [22]. Monte Carlo state generation is especially inefficient [21] in estimating $\overline{P}$ because it randomly generates states from the entire state space without any prior knowledge of state's occurrence probability or performance.

To address the complexity, we device a method for KCP estimation that generates states which have the most impact on KCP in terms of their state probability and performance. In the method, we first find a subset of edges that has the most definite impact on network performance. This subset is then input to *most probable states (MPS) generation method* which yields a KCP estimation. Before providing the details of the method, we first see how we can find the throughput performance of a given state of a mesh network efficiently.

4.3 Throughput Performance of WMN

Many of the previous attempts of throughput capacity evaluation (such as [22], [23], [24], [25], [26]) either give only an asymptotic performance or they are computationally expensive for even a small sized network. In our case, since the throughput has to be calculated for a large number of states, clearly such methods are not useful.

But there is a key difference between the traffic characteristics of general ad-hoc networks and mesh access networks which allows a polynomial time calculation of capacity. Traffic in WMNs always flows between mesh nodes and gateways. This way, traffic aggregation near the gateways is always high which results into traffic bottlenecks near the gateways. The collision domains created around each gateway dictates how much traffic demand can actually be satisfied. *The traffic accumulation around the gateways is so high that throughput reductions due to lower spatial reuse in other regions of the network has only a little effect on the overall capacity.* This was first shown by Knightly et al. in [11], [12] for CSMA case, and was later re-validated for TDMA systems by [27].

Formally, let D bits per second (bps) be the total expected traffic demand of a mesh network. This is the cumulative demand of all mesh nodes in the network. Since every mesh node forwards its data to its gateway, let D_{g_i} denote the traffic demand associated with gateway g_i. Now, let CD_{g_i} be the collision domain of gateway g_i. The collision domain of a gateway is the largest set of links around the gateway such that every pair of links mutually interfere with each other. Let $|\mathrm{CD}_{g_i}|$ denote the size of the collision domain, that is the total traffic of all the links in the collision domain. Now, as described in [11], [27], $|\mathrm{CD}_{g_i}|$ is the total traffic in a collision domain out of which D_{g_i} is the useful traffic. Meaning, during the time proportional to $|\mathrm{CD}_{g_i}|$, the gateway is either transmitting, receiving or deferring for other transmission in the collision domain, while during the time proportional to D_{g_i}, the gateway is either transmitting or receiving (useful communications). This comparison yields a good estimation of how much wireless bandwidth a gateway utilizes for useful transmissions, which in turn determines the throughput capacity. The ratio $\delta_{g_i} = |\mathrm{CD}_{g_i}|/\mathrm{D}_{g_i}$ measures how poorly a gateway utilizes the wireless medium for useful transmission.

Due to high traffic accumulation near the gateways, the ratio δ_{g_i} is almost always larger than one. Now, if the capacity of the gateway g_i is B_{g_i} bps then total aggregate throughput of the mesh network in bps is

$$\mathrm{W} = \sum_{i=1}^{k} \frac{\mathrm{B}_{g_i}}{\delta_{g_i}}. \tag{5}$$

As described in [12], [27], although the throughput calculation presented here assumes a fair MAC, it yields a close approximation of a 802.11 MAC performance.

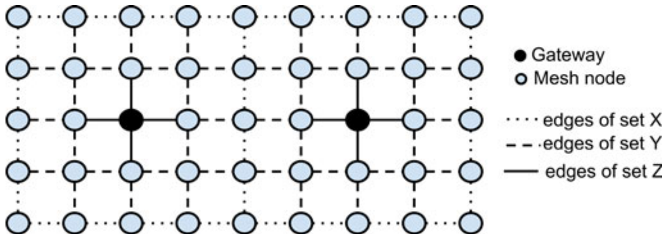


Fig. 5. Edges of sets X, Y and Z.

4.4 Calculating K-Center Performability

The proposed method of estimating KCP consist of two phases. In the first phase, we identify a subset of edges which are most crucial in determining KCP of a WMN. This subset of edges becomes the input for most probable states method in the second phase. The method generates relatively fewer but important states which contribute the most towards KCP value due to their occurrence probability and performance.

4.4.1 Phase 1: Edge Selection

The purpose of edge selection phase is to identify l edges from total of m edges on which the most probable states algorithm can be applied. In the states generated by the MPS algorithm, we fix the states of all edges other than these l edges. It is obvious that criteria for selecting l edges should depend on their expected impact on network throughput and their existence probability. To capture these characteristics of edges, let the edge set E of $G_P(V,E,P)$ be divided into three mutually exclusive and exhaustive subsets as below (also refer to Fig. 5):

- X: an edge $e_i \in X$ if neither of the endpoints of e_i is a gateway and e_i does not interference with any gateway when active.
- Y: an edge $e_i \in Y$ if neither of the endpoints of e_i is a gateway and e_i interferes with a gateway when active.
- Z: an edge $e_i \in Z$ if one of the endpoints of e_i is a gateway and because of that e_i interferes with a gateway when active.

It is obvious that edges of Z will carry more traffic on an average than edges of Y, and edges of Y will carry more traffic than edges of X in general. Since edges of Y and Z belong to a collision domain, their impact on network performance is also more substantial. Based on this understanding we determine the states of edges of each subset and also find l links for MPS algorithm as shown below:

- $\forall e_i \in X, s_{e_i} = 1$. This means that edges of set X are always in operating state in all network states that are generated by MPS algorithm. This is because these edges have the least impact on network performance as we saw in Section 4.3. Forcing these edges to be in operating mode allows maximum influx of traffic from all mesh nodes to gateways.
- $\forall e_i \in Y, s_{e_i} = 1$ if $p_{e_i} \geqslant 1 - p_{e_i}$; $s_{e_i} = 0$ otherwise. This means that edges in set Y are in their most probable state. As we know that these edges have a definite impact on network performance, we ensure that they remain in their most probable state in all network states generated by MPS algorithm, and affect

the KCP evaluation only when they are likely to be in operational mode.

- In terms of KCP, the edges in Z are of foremost importance. Since the network throughput is mostly dependent on these edges, choice of l links for MPS algorithm is made from this set. To do so, first we calculate probability weighted interference score I_{e_i} for every $e_i \in Z$. I_{e_i} is calculated as $I_{e_i} = p_{e_i} \times \Sigma_{e_j \in Z}\, p_{e_j}$, where $e_i, e_j \in Z$, and e_i and e_j mutually interfere. I_{e_i} score of a link shows the likelihood that the link will be in operational mode and it will interfere with how many other links from Z which are also expected to be in their operational mode. We sort the links of Z in decreasing order of their I_{e_i} scores and create a set L using the first l links. These links are then input to MPS algorithm. This way, links which have the most impact on network performance and state occurrence probability are chosen for MPS algorithm. Also, states of the remaining $Z - L$ links are fixed depending on l links. Let $\rho = \max\{p_{e_i}, e_i \in L\}$. Now, for $e_i \in Z - L$, $s_{e_i} = 1$ if $p_{e_i} \geqslant \rho$, and $s_{e_i} = 0$ otherwise.

4.4.2 *Phase 2: Most Probable States Algorithm*

The MPS algorithm presented in [28] sequentially generates the most probable network states in order of decreasing probability. In this work, we utilize this algorithm with l edges found above. Since all other edges in $E - L$ have been assigned a fixed state, MPS algorithm generates states of l edges in each iteration.

Now, for each of the network state generated by MPS algorithm, we find its throughput performance using the method proposed in Section 4.3. The MPS algorithm generates all $\chi = 2^l$ network states where state occurrence probability ($Pr(S)$) of each network state is determined using states of l links in the network state. Let $F(S)$ be the throughput performance of S. The performability estimate $\overline{P}$ is then calculated as $\overline{P} = \Sigma_{S \in \mathcal{X}}(F(S) \cdot \Pr(S))$.

For efficiency, we terminate the state generation when cumulative probability of generated network states reaches 0.99. As described in [28], the actual efficiency of MPS method depends on the average PQ factor. The average PQ factor is defined as the average of n_{e_i}/m_{e_i} ratio for all $e_i \in L$, where n_{e_i} is the probability that e_i is in its least probable state and $m_{e_i} = 1 - n_{e_i}$. Higher value of average PQ factor shows that the difference between link existence and failure probability is lesser for many links, and more number of states are required to be generated for cumulative state probability of 0.99.

5 Analyzing KCA of WMNs

Now we apply the KCA and KCP evaluation methods on different mesh networks. First, to demonstrate their real-world applicability, we apply the KCA evaluation method (CMCS algorithm) to two existing urban mesh networks. For each network, we calculate the link availabilities as shadowing probabilities using their real-world RF profiles.

5.1 KCA of GoogleWiFi

GoogleWiFi [2] is a mesh network deployed in Mountain View, CA which consists of approximately 449 mesh nodes and 59 Internet gateways. For determining link availabilities based on shadow fading, we use the data provided in studies [12]. The studies showed that the network has a path-loss coefficient (η) of 3.5, shadowing factor of (σ) of 8 dB, and reference path loss of 55 dB at 10 meters distance. We choose the receiver sensitivity to be -80 dBm. Given this RF profile and locations of mesh nodes, we apply the KCA evaluation method to GoogleWiFi. The results are presented in Fig. 6a.

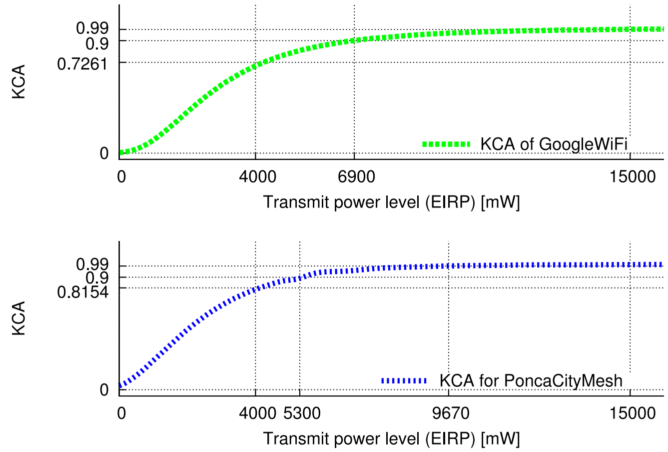


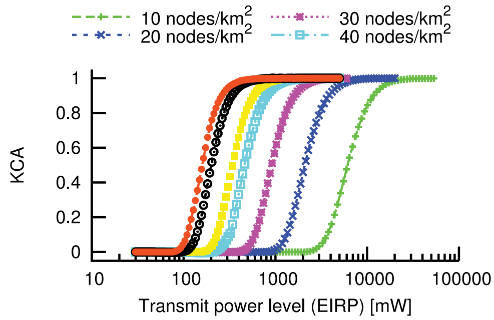
Fig. 6. KCA evaluation of two existing WMNs.

For the purpose of understanding, we increase the transmission power of nodes which should increase the KCA of a WMN. We observe that GoogleWiFi's current node placement requires a very high transmission power from nodes in order to achieve a reasonable KCA. Since maximum EIRP allowed for IEEE 802.11a/b/g standards is 4 W (36 dBm) in USA, we mark this point in the figure to show that GoogleWiFi achieves KCA as low as 0.7261 at that point. Even though transmissions are not allowed at higher transmission powers, we show these values for the purpose of further understanding. *In order to achieve a KCA of 0.9 (often known as "one nine" of network availability) in GoogleWiFi, transmission power as high as 6.9 W is necessary.* This being impractical, we will study the alternative designs to achieve a higher KCA in later sections. As we discussed before, our model does not consider the case where link rate adaptation is used to increase link availability. Since these rate adaptations are used in practice, we expect the actual KCA of GoogleWiFi to be higher than predicted by our model.

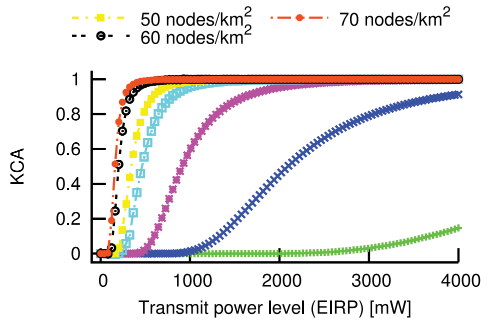
An obvious step at this point is to understand the accuracy of KCA evaluation provided by CMCS method. We defer this discussion until Section 8 where we will show the relationship between bounds and confidence intervals of KCA estimate at each point of simulation. We do so because the accuracy of KCA estimate is sufficiently high (with reasonable efficiency) so that the analysis and general discussion presented here stand true. *Also, each of the data point in KCA and KCP analysis takes upto 7 hours of simulation on average.* Further details of these are provided in Section 8.

5.2 KCA of PoncaCityMesh

We now evaluate the KCA of another urban mesh network named PoncaCityMesh. It is a mesh network deployed in Ponca City, Oklahoma for public Internet access. The network consists of approximately 469 mesh nodes with

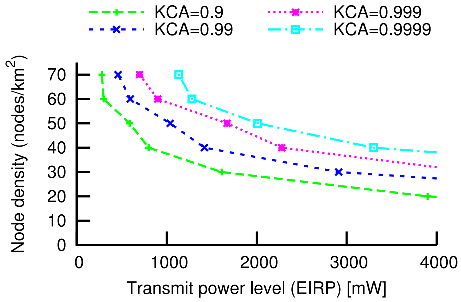


(a) Impact of Tx power and node density on KCA

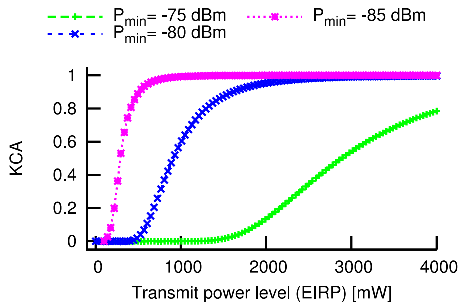


(b) KCA within allowable EIRP

Fig. 7. KCA increase with increase in tx-power is slower in case of low density cases.



(a) Minimum Tx-power/density required for a target KCA



(b) KCA and receiver sensitivity

Fig. 8. As many as 40 nodes per km^2 are necessary to achieve a KCA of four 9s at maximum allowable EIRP.

60 gateways. The RF profile of the city neighborhood is $\eta = 3.4$, $\sigma = 8$ dBm, and reference path loss of 50 dB at 10 meters distance. The KCA evaluation is shown in Fig. 6b. As with GoogleWiFi, PoncaCityMesh also demonstrates a lower KCA. The increase in KCA of PoncaCityMesh as compared to GoogleWiFi can be mostly attributed to a better RF profile of urban neighborhood.

5.3 KCA in Interference-Optimal Node Placement

Since any real-world mesh node placement (such as GoogleWiFi or PoncaCityMesh) is likely to be sub-optimal in terms of interference due to geographical constraints. We now understand KCA for interference-optimal node placement cases. It was previously shown in [12] that a mesh network topology in which mesh nodes (including gateways) are placed in a square grid, and k centers of the network graph are chosen as gateways can achieve the highest availability and performance compared to other model topologies. By using these topological settings, we now show what exact transmission power and/or node density are necessary in order to achieve a high KCA. *It is obvious that increase of transmission power and/or node density will result into higher KCA but we are interested in studying their precise values for any given target KCA.*

Fig. 7a shows how KCA varies with varying density and transmission power of nodes. As expected, it shows that increasingly high transmission power is required to achieve a high KCA when the node density is lower. In all cases with grid node placement, we use RF profile of GoogleWiFi's urban neighborhood in KCA evaluation. With consideration of maximum allowable EIRP, Fig. 7b shows the phase transition behavior of all different densities with maximum EIRP being 4 W.

An optimization problem of designing a mesh network with a certain guaranteed KCA while utilizing minimum resources can be better understood using results of Fig. 8a. It shows for every power level (or node density) what is the minimum necessary node density (or power level) to achieve a targeted KCA. Even at the maximum EIRP of 4 W, it can be observed that number of nodes necessary per km^2 is very high. Also, the required node density value increases with increase in targeted KCA. Guidelines such as the one presented in [29] suggest that current deployment techniques typically uses 8 to 10 nodes per km^2 which is clearly insufficient as per our KCA analysis. Apart from this, real world node placement is unlikely to follow a grid placement, and the added perturbation further decreases the KCA as compared to grid placement. *Our KCA analysis shows that a grid network with 20 nodes per* km^2 *with each node operating at maximum EIRP yields a network in which mesh nodes are connected to gateways only 90 percent of the time. To increase the availability upto 0.9999 as many as 40 nodes per* km^2 *is necessary. The value is likely to be even higher in real world deployments that can not follow a grid placement.*

One of the parameters which we did not vary previously in KCA evaluation is receiver sensitivity. On one hand, a higher receiver sensitivity is necessary for links to operate at a higher physical layer data, while on the other hand, increase of receiver sensitivity results into decrease of communication range of a node which in turn reduces the KCA value. This is demonstrated in Fig. 8b where we vary receiver sensitivity of nodes while fixing the node density to 30 nodes per km^2. It is observed that phase transition widths increase with the increase in receiver sensitivity, and a much higher transmission power or node density is necessary if we want to operate the links of a mesh network at higher link rates.

The requirement of higher transmission power in order to design a network with lower node density or higher node density but higher link rates come at a penalty of increased interference. In general, the increased transmission power reduces the throughput performance of a network due to increased interference. This way, KCA and network performance can be at tradeoff and it might be very difficult to design a highly performable network with high KCA value. We try to understand this by exploring KCP of WMN next.

6 Analyzing KCP of WMNs

Increasing transmission power and/or node density increases KCA but can reduce the throughput performance due to increased interference. We analyze this tradeoff next by understanding the relationship between KCP and the deployment factors.

6.1 KCP in Interference-Optimal Node Placement

We first analyze the KCP of WMNs where mesh nodes are placed in a square grid, and then evaluate the KCP of GoogleWiFi and PoncaCityMesh. In all cases, a total of 300 Mbps traffic is inserted into the network which is uniformly divided among the mesh nodes.

6.1.1 Comparing KCP with Performance of Network Simulations

First, we compare the KCP value calculated by our technique with the performance of network simulator (NS-3 [30]). This ensures that our KCP analysis is valid even when actual 802.11 PHY/MAC are used along with other higher layer protocols. Although we find that performing such a comparison is extremely difficult because performance of the network simulator with different link availability values can not converge even after a very long period of time. An alternate approach is to generate all possible states of the network and run the network simulation for each of the states. This allows us to get the performance of each state with a smaller convergence time but this approach can not scale because of extremely large number of possible network states. In fact, this is our central motivation behind the design of KCP evaluation technique in this work because empirical evaluation of KCP is extremely challenging, and it takes a very long time to take even a moderate number of measurements to estimate the KCP in practice.

In order to verify that our KCP evaluation is comparable with the performance of real networks, we run NS-3 simulations for *the most probable state* of the network. As we discussed in Section 4.4.2, in the most probable state of the network, all the links operate in their most probable state as well. We input this network state in our NS-3 simulations. We fix the node density to be 40 nodes per km^2 with 170 mesh nodes and 16 gateways. We use 802.11a PHY/MAC with receiver sensitivity of -80 dBm with fixed link data rate of 9 Mbps (based on 802.11a). A total of 300 Mbps of UDP input traffic is equally divided among the mesh nodes.

Fig. 9 shows the network throughput of the most probable network state. We also show the KCP estimate of the same network configuration for comparison. Simulations are ran for ten instances and their average values are plotted

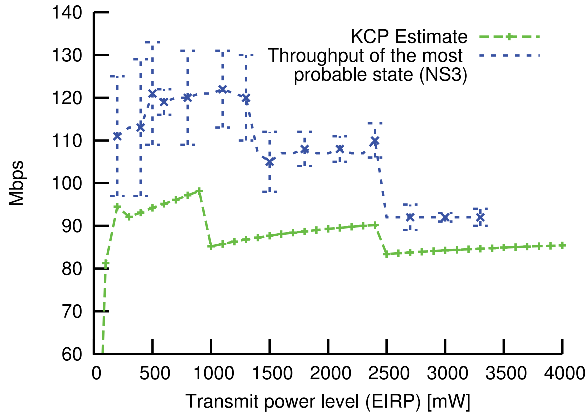


Fig. 9. Comparison between KCP and throughput performance of the most probable network state (NS-3).

with 90 percent confidence intervals. As expected, the throughput of the most probable state is higher than the KCP value because result of NS-3 run is an absolute value of throughput while KCP is availability-weighted throughput across all possible states. This shows that utilizing absolute value of throughput of the most probable state of the network is very optimistic in estimating the true quality of the network. The NS-3 simulation verifies that our estimation of network throughput is indeed comparable to performance of network simulators. They both follow the same pattern of increase and decrease which also confirms that overlap between the collision domains of gateways dictates how performance changes with varying transmission power. We next characterize KCP and provide details of its behavior.

6.1.2 Characteristics of KCP

Fig. 10 shows the how KCP changes with varying node density and transmission power. The main observations are as follows:

(1) In general, for every node density, KCP first increases with increase in transmission power. This is due to the fact that availability of mesh nodes increases in this initial period which allows more and more nodes to transfer their data to gateways. This increase in throughput performance yields an increasing KCP initially. Further increase in transmission power results into decrease of KCP because of increasing interference among the nodes which reduces the throughput performance. *Here, availability of mesh nodes is still higher but the drastic decrease in the performance results into lower KCP.*

(2) *KCP shows a saw-tooth wave like pattern with increase in transmission power.* This is because throughput performance of a WMN displays a behavior similar to a step function where the performance decreases sharply and then remains unchanged until another subsequent decrease. The sharp decrease is due to fact that at a certain transmission power level, interference collision domains of nearby gateways start overlapping with each other. The performance remains unchanged until the subsequent overlap of collision domains. During this phase where throughput performance is unchanged, the availability still increases with increase in transmission power. This results into increase of their joint metric—KCP in this region with saw-tooth-like overall behavior.

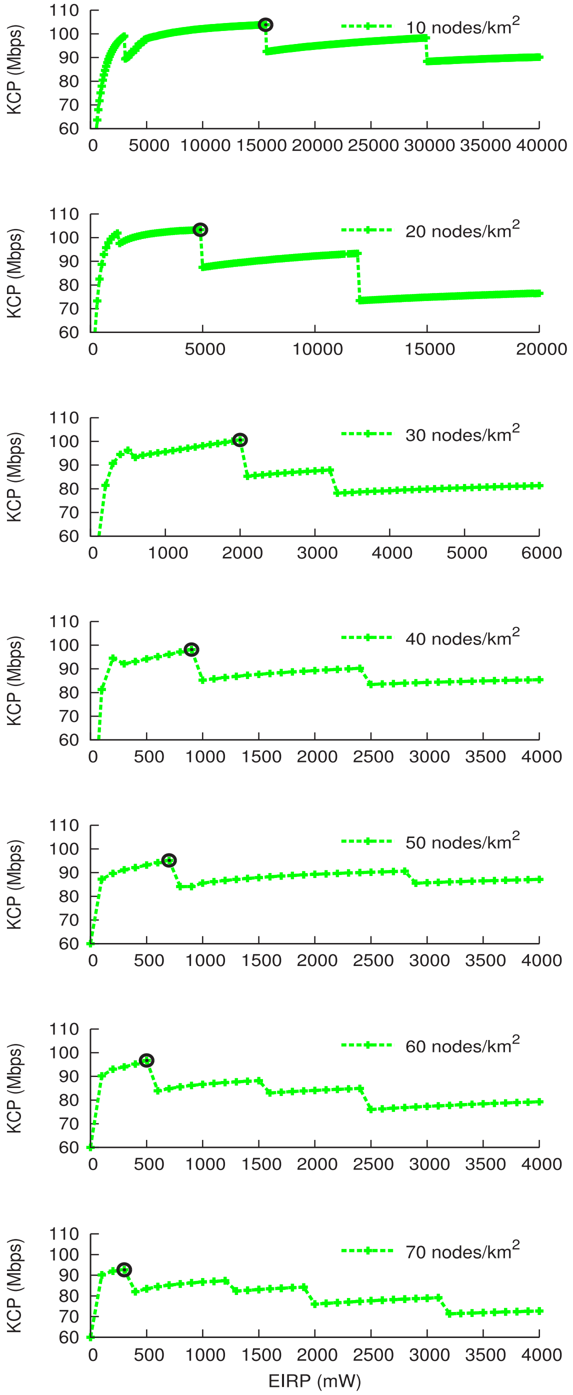


Fig. 10. Impact of node density and transmission power on network KCP.

(3) Note that *decrease in KCP is much faster for higher node densities*. This is due to the fact that the diminishing effect of interference is much worse at higher density, and throughput performance starts dropping at lower power levels yielding lower KCP value.

(4) A interesting observation is that *absolute values of maximum KCP remain comparable even at very high density values*. This is because higher density yields a lower performance due to interference but the availability with which it is delivered is higher for them. This is an especially important result because it differs from the conventional understanding that higher node density yields lower network

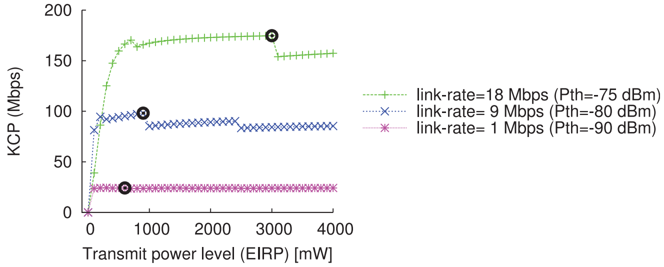


Fig. 11. Impact of receiver sensitivity on KCP.

throughput [23]. When the design strategy shifts from performance to performability, in fact higher node density can be beneficial.

(5) Note that KCP does not become zero even at very low transmission power. At this point, almost all mesh nodes are disconnected from gateways and are unable to transfer any data. But since the *gateways also act as access point to clients, their data gets through Internet even if no other mesh nodes are connected.*

For a given receiver sensitivity, links operate at a specific data rate in absence of any interference. Increasing receiver sensitivity increases link data rate and network performance but reduces the network availability. This motivates us to study the impact of receiver sensitivity on KCP. As the first step towards studying performance-degradable network, we assume that the links are not performance-degradable. This means that a link provides a specific data rate only when sensitivity requirement is met, and is denoted dysfunctional otherwise, even though it can provide a lower data rate in its degraded state. We leave this case of degradable links for future study. Here, we fix the node density to be 15 nodes per km^2 and vary the transmission power and receiver sensitivity. We use the parameters of 802.11a for mapping receiver sensitivity to its corresponding data rate. Fig. 11 shows that in fact increasing receiver sensitivity results into increased KCP due to sharper increase in performance but not so sharp decrease in availability.

6.2 KCP of GoogleWiFi and PoncaCityMesh

We now evaluate KCP for GoogleWiFi and PoncaCityMesh network. The results are shown in Figs. 12a and 12b respectively. The general behavior discussed in case of grid topology remains more or less similar for both the networks. One

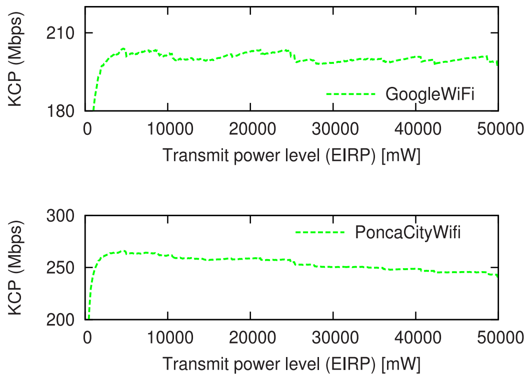


Fig. 12. KCP of (a) GoogleWiFi (b) PoncaCityMesh.

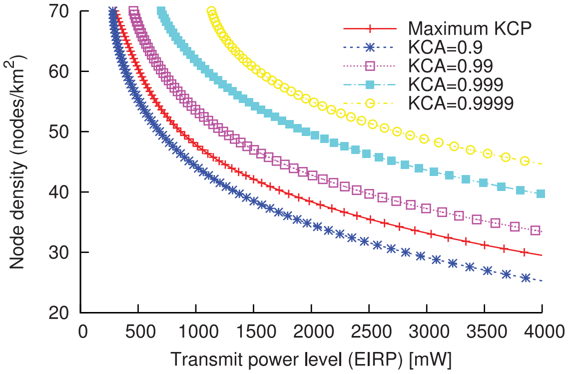


Fig. 13. Tx-power and/or density necessary to achieve a given KCA or KCP.

of the reason PoncaCityMesh achieves better KCP is because it contains more number of gateways compared to GoogleWiFi.

7 KCA-KCP Tradeoff

The central objective of this work is to study KCA and KCP of WMNs so that we can gain better insights on how to design mesh networks that can guarantee a certain level of KCA and KCP. Both of these metrics can be used by network designers to establish useful SLAs with service providers.

A surprising result that we observe here is that it is not possible to maximize KCA and KCP simultaneously. It can be observed from Figs. 7a and 10 that for any given node density, increasing transmission power increases KCA and KCP upto a certain point. Beyond this, increase of transmission power may or may not increase KCA but starts decreasing KCP. This trade-off shows that in fact it is challenging to choose a tx-power level that can maximize both KCA and KCP.

For further understanding, we plot the data of Fig. 8a (with curve smoothing applied) along with the data extracted from Fig. 10 which shows the power level at which each density achieves the maximum KCP. This data is shown in Fig. 13. We can observe that *in fact a mesh network achieves maximum KCP when its KCA is between 0.9 to 0.99*. Since these results are obtained for interference-optimal node placement, any further perturbation will further increase the minimum necessary transmission power for a target KCA. This makes the KCA-KCP tradeoff even worse because the difference between power levels at which maximum KCA and KCP are obtained increases.

This KCA-KCP tradeoff shows that a network designer can not optimize a mesh network for both KCA and KCP, and a clear choice has to be made between them. Traditionally, since only average-case throughput has been considered in design, availability and performance were not at odds. This was proven by many previous seminal works [4], [5]. As the design focus shifts from performance to performability, intelligent interference management techniques are necessary to maintain higher performance at higher interference.

Other important observation is that at lower densities (for 10, 20, 30 and 40 cases) minimum power necessary for higher KCP occurs after first sharp decrease of KCP

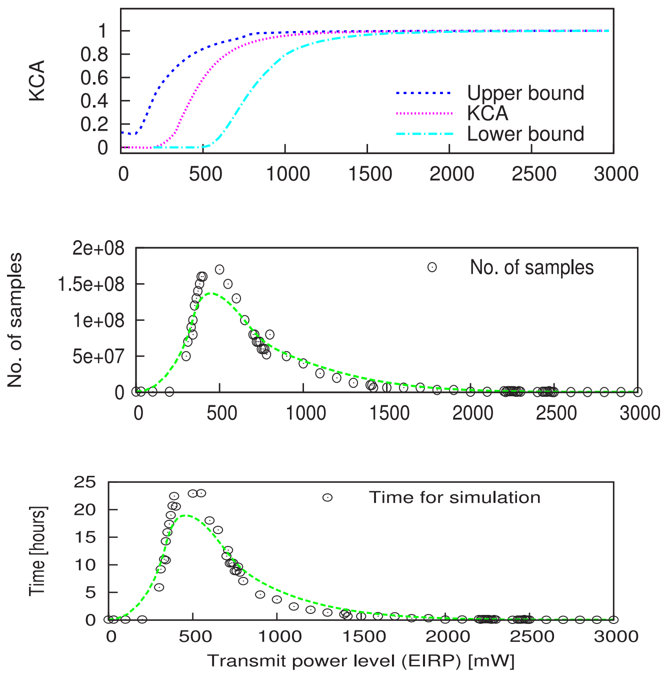


Fig. 14. Efficiency of CMCS method: (a) Edge-packing bounds, (b) number of samples required, and (c) simulation time.

(Fig. 10). The sharp decrease occurs due to overlapping of collision domains of nearby gateways at high power levels. *This means that in fact it is advisable to operate the network with overlapping collision domains of gateways because the availability benefits of doing this is more compared to performance penalties associated with it.*

8 Efficiency and Accuracy Analysis of KCA and KCP Evaluation

8.1 KCA—Accuracy and Efficiency

Now we validate our claims about accuracy and efficiency of KCA evaluation method. First, Fig. 14a shows the nature of edge packing bounds along with the actual value of KCA estimated by CMCS method. Fig. 14 shows the results for node density of 40 nodes per km^2. The bounds indicate the amount of reduction in search space when executing the Monte Carlo simulation. This is reflected in Figs. 14b and 14c which show the number of samples necessary and time taken for KCA estimation respectively. The simulations were run on a desktop computer with 1.6 GHz processor and 1 gigabytes of memory. As expected, simulation time and number of samples are a function of bounds as well as number of edges in the networks.

It is obvious that the efficiency of CMCS procedure also depends on the size of the network (node density and number of edges). To understand this, we normalize the required time of simulation using the number of edges in the network. The results are shown in Fig. 15. The transmission power level at which the simulation time is the maximum increases with decrease in density. This is because the phase transition occurs at higher power levels for lower density.

Lastly, we evaluate the efficiency of KCA estimation method. We set the accuracy level of CMCS method to be 10^{-5}. Fig. 16 shows the relation between (upper

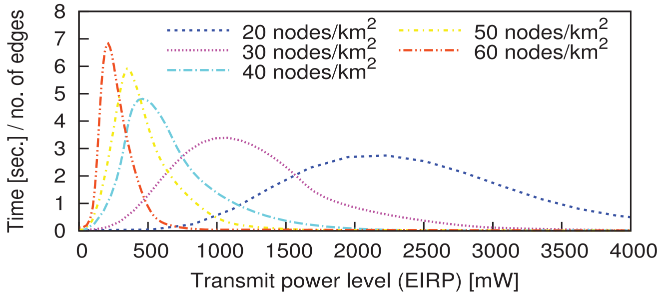


Fig. 15. Time required for KCA estimation in different network densities.

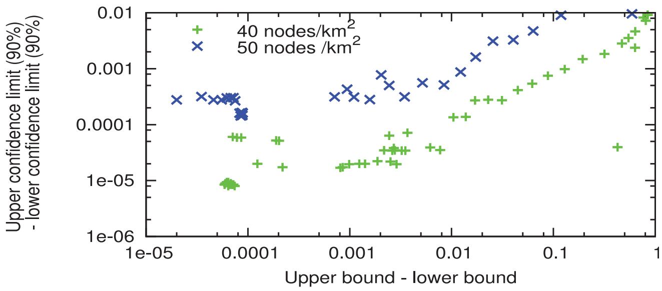


Fig. 16. Accuracy of KCA estimation method.

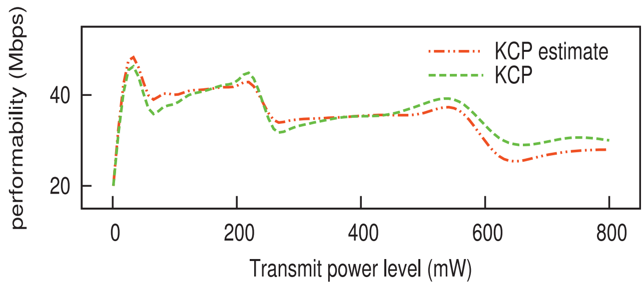


Fig. 17. Accuracy of KCP estimation method.

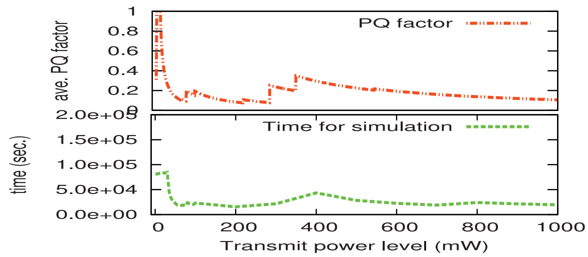


Fig. 18. Efficiency of KCP estimation.

bound-lower bound) and (upper confidence limit-lower confidence limit). The confidence intervals are calculated using the method suggested in [16]. As the bounds become tighter, the corresponding confidence intervals also shrink.

8.2 KCP—Accuracy and Efficiency

To examine how accurate is the proposed KCP evaluation method, we apply it on a smaller network with 20 nodes, 40 links and 100 Mbps of input traffic. We generate all ($2^{40} \approx 10^{12}$) network states and calculate the exact value of $\overline{P}$. We also apply the proposed KCP estimation method with $l = 20$. The results are shown in Fig. 17. It can be observed that the proposed method yields a good estimate of the actual KCP value. Also, the time requirement of proposed method is significantly lesser (nearly 26 hours in worst case) than entire state space generation (approx. 198 hours in worst case). Since the presented comparison is only for a small network, we don't claim that the same holds for larger networks. We are unaware of any better method of KCP evaluation that can be used for comparison, which we consider to be one of the limitations of our work.

We now show how efficient is the proposed KCP evaluation method. As mentioned before, time requirement of MPS algorithm depends the average PQ factor. For node density of 60 nodes per km^2, Fig. 18 shows the average PQ factor and the time required for KCP estimation. Higher value of average PQ factor shows that more network states are required to be generated for a cumulative state probability of 0.99.

9 Conclusions

In this paper, we defined connection and performance robustness for wireless mesh networks. We developed efficient methods that can estimate both KCA and KCP for large urban-scale mesh networks. We apply our evaluation methods to two existing mesh networks to demonstrate that their current design can not guarantee a reasonable level of availability or performability. Using hundreds of hours of simulations, we showed that when performability is considered as opposed to average case throughput performance, there does not exist a transmission power or node density that can maximize both availability and performability. We characterized the nature of availability and performability, and outlined new design policies that can be followed in order to design mesh networks that have high KCA and KCP.

References

[1] A. Van Moorsel, "Metrics for the internet age: Quality of experience and quality of business," in *Proc. 5th Int. Workshop Performability Model. Comput. Commun. Syst.*, Arbeitsberichte des Inst. für Informatik, Universität Erlangen-Nürnberg, Germany, 2001, vol. 34, no. 13, pp. 26–31.

[2] Google wifi, Mountain View, CA, USA. (2008). [Online]. Available: http://wifi.google.com/

[3] Ponca city wifi. (2009). [Online]. Available: http://www.myponcacity.com/wifi/poncaradios.html

[4] V. Kawadia and P. Kumar, "Principles and protocols for power control in wireless ad hoc networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 1, pp. 76–88, Jan. 2005.

[5] J. A. Fuemmeler, N. H. Vaidya, and V. V. Veeravalli, "Selecting transmit powers and carrier sense thresholds in csma protocols for wireless ad hoc networks," in *Proc. 2nd Annu. Int. Workshop Wireless Internet*, 2006, pp. 15–21.

[6] Okmulgee, Oklahoma—municiple wireless. (2011). [Online]. Available: http://www.muniwireless.com/2011/06/16/okmulgee-oklahoma-deploys-municipal-wifi-network

[7] Kansas city—smart grid project. (2011). [Online]. Available: http://www.tropos.com/news/pressreleases/7_27_2011.php

[8] A. Annamalai, C. Tellambura, and V. Bhargava, "Simple and accurate methods for outage analysis in cellular mobile radio systems-A unified approach," *IEEE Trans. Commun.*, vol. 49, no. 2, pp. 303–316, Feb. 2001.

[9] A. Snow, U. Varshney, and A. Malloy, "Reliability and survivability of wireless and mobile networks," *Comput.*, vol. 33, no. 7, pp. 49–55, 2000.

[10] J. Robinson, R. Swaminathan, and E. W. Knightly, "Assessment of urban-scale wireless networks with a small number of measurements," in *Proc. 14th ACM Int. Conf. Mobile Comput. Netw.*, 2008, pp. 187–198.

[11] J. Robinson, M. Uysal, R. Swaminathan, and E. Knightly, "Adding capacity points to a wireless mesh network using local search," in *Proc. IEEE 27th Conf. Comput. Commun.*, 2008.

[12] J. Robinson and E. Knightly, "A performance study of deployment factors in wireless mesh networks," in *Proc. 26th IEEE Int. Conf. Comput. Commun.*, 2007, pp. 2054–2062.

[13] C. Bettstetter and C. Hartmann, "Connectivity of wireless multihop networks in a shadow fading environment," *Wireless Netw.*, vol. 11, no. 5, pp. 571–579, Sep. 2005.

[14] F. Xing and W. Wang, "On the critical phase transition time of wireless multi-hop networks with random failures," in *Proc. 14th ACM Int. Conf. Mobile Comput. Netw.*, 2008, pp. 175–186.

[15] R. Hekmat and P. Van Mieghem, "Connectivity in wireless ad-hoc networks with a log-normal radio model," *Mobile Netw. Appl.*, vol. 11, no. 3, pp. 351–360, Jun. 2006.

[16] G. S. Fishman, "A monte carlo sampling plan for estimating network reliability," *Oper. Res.*, vol. 34, no. 4, pp. 581–594, Jul. 1986.

[17] C. J. Colbourn, *The Combinatorics of Network Reliability*. New York, NY, USA: Oxford Univ. Press, 1987.

[18] W. T. Tutte, "On the problem of decomposing a graph into n connected factors," *J. London Math. Soc.*, vol. 36, pp. 221–230, 1961.

[19] J. Roskind and R. Tarjan, "A note on finding minimum-cost edge-disjoint spanning trees," *Math. Oper. Res.*, vol. 10, pp. 701–708, 1985.

[20] E. Manzi, M. Labbe, G. Latouche, and F. Maffioli, "Fishman's sampling plan for computing network reliability," *IEEE Trans. Rel.*, vol. 50, no. 1, pp. 41–46, Mar. 2001.

[21] D. D. Harms, *Network Reliability: Experiments with a Symbolic Algebra Environment*, M. Kraetzl, Ed. Boca Raton, FL, USA: CRC Press, 1995.

[22] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," in *Proc. 9th Annu. Int. Conf. Mobile Comput. Netw.*, 2003, pp. 66–80.

[23] P. Gupta and P. Kumar, "The capacity of wireless networks," *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 388–404, Mar. 2000.

[24] G. Brar, D. M. Blough, and P. Santi, "Computationally efficient scheduling with the physical interference model for throughput improvement in wireless mesh networks," in *Proc. 12th Annu. Int. Conf. Mobile Comput. Netw.*, 2006, pp. 2–13.

[25] M. Gastpar and M. Vetterli, "On the capacity of wireless networks: the relay case," in *Proc. IEEE 21st Annu. Joint Conf. Comput. Commun. Soc.*, 2002. vol. 3, pp. 1577–1586.

[26] J. Jun and M. Sichitiu, "The nominal capacity of wireless mesh networks," *IEEE Wireless Commun.*, vol. 10, no. 5, pp. 8–14, Oct. 2003.

[27] P. Pathak and R. Dutta, "Impact of power control on capacity of tdm-scheduled wireless mesh networks," in *Proc. IEEE Int. Conf. Commun.*, 2011, pp. 1–6.

[28] T. M. S. Gomes and J. M. F. Craveirinha, "Algorithm for sequential generation of states in failure-prone communication network," *Proc. IEE Commun.*, vol. 145, no. 2, pp. 73–79, Apr. 1998.

[29] "Five keys to successful metro-scale wi-fi deployment," *tropos Netw. White Paper*, Dec. 2004.

[30] Network Simulator. (2013). [Online]. Available: http://www.nsnam.org/

Parth H. Pathak received the doctoral degree in computer science at North Carolina State University, Raleigh, in 2012. He is a postdoctoral scholar in the Computer Science Department at the University of California, Davis. His current research interests include design and development of next-generation wireless networks, network analytics, and mobile computing. In the past, he has worked on cross-layer protocol design and optimization of wireless networks.

Rudra Dutta received the BE degree in electrical engineering from Jadavpur University, Kolkata, India, in 1991, the ME degree in systems science and automation from the Indian Institute of Science, Bangalore, India, in 1993, and the PhD degree in computer science from North Carolina State University, Raleigh, in 2001. He has been employed from 2001 to 2007 as an assistant professor, from 2007 to 2013 as an associate professor, and since 2013 as a professor, in the Department of Computer Science at the North Carolina State University, Raleigh. His current research interests include design and performance optimization of large networking systems, Internet architecture, wireless networks, and network analytics. His research is supported currently by grants from the US National Science Foundation (NSF), the National Security Agency, and industry, including a recent GENI grant and a FIA grant from NSF. He has was the program chair for the Optical Networking Symposium at IEEE Globecom 2008, general chair of IEEE ANTS 2010, and as a guest editor of a special issue on Green Networking and Communications of the *Elsevier Journal of Optical Switching and Networking*. He is currently on the Steering Committee of IEEE ANTS 2013, and on the editorial board of the *Elsevier Journal of Optical Switching and Networking*.

Prasant Mohapatra received the doctoral degree from Penn State University in 1993. He is a professor in the Department of Computer Science at the University of California, Davis (UC Davis). He is currently the interim vice-provost and the CIO of UC Davis. He was the department chair of computer science during 2007 to 2013, and held the Tim Bucher family endowed chair professorship during that period. In the past, he has been on the faculty at Iowa State University and Michigan State University. He is the editor-in-chief of the *IEEE Transactions on Mobile Computing*. He was on the editorial board of the *IEEE Transactions on Computers*, *IEEE Transactions on Mobile Computing*, *IEEE Transaction on Parallel and Distributed Systems*, *ACM Journal of Wireless Networks*, and *Ad Hoc Networks*. He was the program chair and the general chair and has been on the program/organizational committees of several international conferences. His research interests include the areas of wireless networks, mobile communications, cybersecurity, and Internet protocols. He received an Outstanding Engineering Alumni Award in 2008.