

A Novel Family of Geometric Planar Graphs for Wireless Ad Hoc Networks

Xu Li, Nathalie Mitton, Isabelle Simplot-Ryl, David Simplot-Ryl

► **To cite this version:**

Xu Li, Nathalie Mitton, Isabelle Simplot-Ryl, David Simplot-Ryl. A Novel Family of Geometric Planar Graphs for Wireless Ad Hoc Networks. Proc. 30th IEEE International Conference on Computer Communications (IEEE INFOCOM 2011), Apr 2011, Shanghai, China. 2011. <inria-00587873>

HAL Id: inria-00587873

<https://hal.inria.fr/inria-00587873>

Submitted on 21 Apr 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Novel Family of Geometric Planar Graphs for Wireless Ad Hoc Networks

Xu Li^{†‡}, Nathalie Mitton[†], Isabelle Simplot-Ryl[†], and David Simplot-Ryl[†]

[†] INRIA/CNRS/Univ. Lille 1, France, {firstname.lastname}@inria.fr

[‡] ECE, University of Waterloo, Canada

Abstract—We propose a radically new family of geometric graphs, *i.e.*, *Hypocomb*, *Reduced Hypocomb* and *Local Hypocomb*. The first two are extracted from a complete graph; the last is extracted from a Unit Disk Graph (UDG). We analytically study their properties including connectivity, planarity and degree bound. All these graphs are connected (provided the original graph is connected) planar. Hypocomb has unbounded degree while Reduced Hypocomb and Local Hypocomb have maximum degree 6 and 8, respectively. To our knowledge, Local Hypocomb is the first strictly-localized, degree-bounded planar graph computed using merely 1-hop neighbor position information. We present a construction algorithm for these graphs and analyze its time complexity. Hypocomb family graphs are promising for wireless ad hoc networking. We report our numerical results on their average degree and their impact on FACE routing [2]. We discuss their potential applications and some open problems.

I. INTRODUCTION

A planar graph is a sparse graph where edges intersect only at their end vertices. It has been widely adopted in different domains to solve various problems, *e.g.*, circuit layout design on computer chips, image segmentation in computer vision, facility layout design in operations research, just to mention a few. In these applications, the position of all vertices is known, and edges can be added between any two vertices. Planarization is equivalent to an edge removal process on a complete graph with connectivity preservation. In some other cases, edge addition is subject to distance constraint, bringing about the problem of planarization on a Unit Disk Graph (UDG). In the sequel, we always assume connected UDG, and two intersecting (or crossover) edges imply that the two edges intersect, but not at their end vertices.

Define the unit circle $C_\gamma(a)$ of a vertex a as the circle of radius equal to a unit distance γ and centered at a . The unit disk $D_\gamma(a)$ of a is the area enclosed by $C_\gamma(a)$. In UDG, there is an edge between two vertices a and b and they are said ‘adjacent to’ or ‘neighboring’ each other if and only if $b \in D_\gamma(a)$ (equivalently, $a \in D_\gamma(b)$). We denote by $V_{NBR}(a)$ the closed neighborhood (neighbor set) of a (including a) and by $V_{NBR}(a, b)$ the closed common neighborhood of a and b .

Wireless ad hoc networks (*e.g.*, sensor networks) where nodes have the same maximum transmission range γ (unit distance) are commonly modeled as UDG. In such networks, each node is static and assumed to know its own geographic position by attached GPS device or some other means. Two nodes are neighbors (*i.e.*, have an edge in between) if and only if they are within each other’s transmission range (*i.e.*, unit disk). Periodic ‘hello’ message is a basic ad hoc networking

technique for neighborhood discovery [6]. By this technique, each node is able to gather the location information of all neighboring nodes. In the past decade, several well-known position-based ad hoc routing protocols [4] were proposed. They all rely on planar network topology for guaranteeing packet delivery. In general, UDG is not planar. A planar subgraph has to be extracted through a planarization procedure.

In wireless networks, nodes share the communication media and have limited channel capacity. The main communication cost is therefore message transmissions. To minimize the control overhead on the network, graph planarization ought to be carried out in a distributed fashion without resorting to any global knowledge and with a minimal total number of message transmissions per wireless node. Ideally, it involves no message transmission in addition to the built-in ‘hello’ message. Packets have constant size at MAC layer. Transmission of a long message requires message fragmentation and leads to increased number of transmissions. Long message consumes more transmission power than short message, are more likely to cause error and should be avoided. Thus as an additional requirement, no modification should be made to the default ‘hello’ message (normally containing constant-sized information such as sender position) during planarization. In summary, graph planarization in wireless ad hoc networks is expected to be a *strictly localized* procedure, where each node makes consistent planarization decision independently using 1-hop neighborhood information only.

There are a few strictly localized planar graphs such as GG, RNG [7] and PDel [16] and a few non-strictly localized planar graphs such as LMST [15] and LDel [5], [13]. The *degree* $\Delta(G)$ of a graph G is the maximum node degree in the graph. It is often desirable that $\Delta(G)$ is small and bounded above by a constant. In wireless communications, a small node degree reduces the contention and interference and helps to mitigate the hidden and exposed terminal problems at MAC layer. In bluetooth scatternets, each node is required to have maximum degree 7. All the above local planar graphs but LMST have unbounded degree in nature, while LMST construction is not strictly localized (requires 2-hop information). Li et al. [12] modified RNG construction such that the degree is limited to a small constant. However, the modification requires each vertex to be associated a unique identifier (ID), which does not necessarily exist in, *e.g.*, sensor networks. Li et al. [16] showed that degree can be limited to a constant with connectivity and planarity preservation by applying Yao structure [19].

In this paper, we propose a radically new family of geometric planar graphs, completely different from any known graph, and focus on their theoretical properties. We first introduce *Hypocomb* (*Hypotenuse-comb*), which is the ‘dual’ (an abused use of term duality) of a truncated mesh [11] referred to as *Besh* (*Blocked-mesh*). Given a set of vertices in the Euclidean plane, Besh is constructed by drawing rays synchronously from each vertex in four directions and allowing distance-based blocking when they meet each other. Hypocomb is obtained by linking vertices that have a ray-blocking relation in Besh. We prove that Hypocomb is connected planar with unbounded degree. Then we propose to reduce its degree to 6 by applying constrained edge creation rule, without jeopardizing its connectivity and planarity: link two vertices if and only if they have a mutual ray-blocking relation. The resultant Hypocomb is called *Reduced Hypocomb*. After that, we present *Local Hypocomb* on the basis of UDG. It is constructed in a strictly localized manner, by removing any UDG edge that does not belong to the Reduced Hypocomb of the closed common neighborhood of its end vertices. We prove that Local Hypocomb remains connected planar and has slightly larger degree 8. Local Hypocomb is the first strictly-localized, degree-bounded planar graph computable using 1-hop neighbor position information only. It may serve as alternative graph in geographic routing for providing delivery guarantee in wireless ad hoc networks. We present, along with complexity analysis, a construction algorithm for Hypocomb family graphs. Through simulation we study their average degree and their impact on the well-know FACE routing protocol [2], in comparison with widely-adopted *Delaunay triangulation* and *Gabriel Graph*. Simulation results imply that Local Hypocomb is superior to Gabriel Graph. We indicate that Hypocomb and Reduced Hypocomb may be built in a localized way among actor nodes in emerging wireless sensor and actor networks and provide a generic solution to the challenging actor-actor coordination problem.

The rest of this paper is organized as follows. We review existing local planar graphs in Sec. II. We propose and analyze Hypocomb family graphs in Sec. III - V, along with numeric results being reported in Sec. VI. We conclude the paper by describing some of their potential applications and open problems for future research in Sec. VII.

II. RELATED WORK

There is only a few localized planar graphs in the literature. Given a vertex set V in the Euclidean plane, in the following we will briefly introduce how to construct these graphs. The containment relations among these graphs are given below.

$$\text{MST} \subseteq \begin{matrix} \text{LMST} \\ \text{RNG}' \end{matrix} \subseteq \text{RNG} \subseteq \text{GG} \subseteq \begin{matrix} \text{PDel} \\ \text{LDel} \end{matrix} \subseteq \text{Del}$$

A *Minimum Spanning Tree* (MST) is a subgraph connecting all the vertices with weighted edges that lead to minimum total weight. If edges are weighted by the Euclidean distance of their end vertices (as in our context here), it is called Euclidean MST, and it has degree bounded above by 6 [17]. In general, V may have many MST unless each edge has a

unique weight. MST can not be computed locally, *i.e.*, each node can not determine which edges are in MST by purely using the information of the nodes within some constant hops [12]. MST is not spanner, *i.e.*, having no constant spanning ratio. The *spanning ratio* of a graph is the maximum ratio of the Euclidean length of the shortest path connecting two arbitrary vertices and their direct Euclidean distance.

A *Local MST* (LMST) [9] is a connected subgraph of UDG, constructed locally using 2-hop neighborhood information as follows: at each vertex u , compute the MST of the sub-graph of $V_{NBR}(u)$; add incident edge uw to LMST if and only if the edge is in both $\text{MST}(V_{NBR}(u))$ and $\text{MST}(V_{NBR}(w))$. LMST contains MST as subgraph and has the same degree bound 6. In [15], it is proved that LMST is also planar, and the notion is extended to k -Local MST (LMST $_k$) with k -hop neighborhood information being used. LMST is not spanner.

Gabriel Graph (GG) is built by connecting any two vertices u and w if and only if the closed disk $\text{disk}(u, w)$ with uw as diameter contains no other vertex from V , while *Relative Neighborhood Graph* (RNG) is built by connecting u and w if and only if the interior of their lune $\text{lune}(u, w)$ (*i.e.*, the intersection of the two circles of radius $|uw|$ centered at u and w) contains no other vertex. Both GG and RNG are connected planar if the original graph is UDG. Each of them can be constructed strictly locally by each vertex checking the construction condition for its neighbors only. GG and RNG are so-called proximity graphs [7]. Neither of them has constant bounded spanning ratio or bounded degree. A study of their spanning ratio was presented in [1].

Assuming each vertex is associated with a unique ID, a modified RNG, called RNG', was proposed in [12]. RNG' contains all edges uw such that the interior of $\text{lune}(u, w)$ contains no vertex, and (2) there is no vertex v on the boundary of $\text{lune}(u, w)$ such that $ID(v) < ID(w)$ and $|vw| < |uw|$, and (3) there is no vertex v on the boundary of $\text{lune}(u, w)$ such that $ID(v) < ID(u)$ and $|vu| < |uw|$, and (4) there is no vertex v on the boundary of $\text{lune}(u, w)$ such that $ID(v) < ID(u)$, $ID(v) < ID(w)$, and $|vu| = |uw|$. RNG' is a subgraph of RNG. It is proved that RNG' has maximum degree 6 and contains MST as subgraph.

A *Delaunay triangulation* (Del) is built by connecting any two vertices $u, w \in V$ if and only if the circumcircle of the triangle defined by u, w and any other vertex $v \in V$ is empty. Given V , there may be more than one Delaunay triangulation, but only if V contains four or more co-circular vertices. Del has constant spanning ratio [8]. Del can not be constructed locally, because it may contain arbitrary long edges.

A connected planar was proposed for UDG on the basis of Del and under the assumption of no four co-circular vertices in [16]. The graph is a subset of Del and thus named *Partial Delaunay triangulation* (PDel). It contains only a few more edges than GG. To construct PDel, each node u for each $w \in V_{NBR}(u)$ checks the following conditions: (1) $\text{disk}(u, w)$ is empty (*i.e.*, uw belongs to GG); (2) $\text{disk}(u, w)$ contains vertices only on one side of uw , with x being one of those vertices that maximizes $\angle uxw$ in triangle Δuxw such that

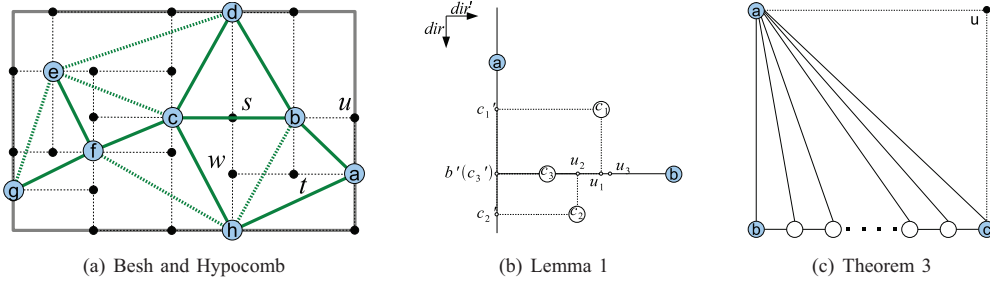


Fig. 1. Illustrations for Besh and Hypocomb

$\angle uxw + \angle uyw < \pi$, where $\angle uyw$ is in triangle Δuyw and maximum with y being from a subset of vertices (referred to as search set) on the other side of uw . The search set can be defined either as the set of common neighbors of u and w (1-hop knowledge suffices for planarization in this case) or as the 2-hop neighbor set of u . If any of these two conditions holds, edge uw is added to PDel. PDel has unbounded degree. Its degree is limited to 7 after Yao structure [19] is applied. PDel has no constant bounded spanning ratio.

Another Del-based connected planar graph, called *Localized Delaunay triangulation* (LDel), was proposed for UDG independently, in [5] and [13]. As PDel, it contains GG as subgraph; unlike PDel, it has good spanning ratio. The planarization process works as follows: $\forall u \in V$, compute $Del(V_{NBR}(u))$; $\forall w \in V_{NBR}(u)$, uw is added to LDel if $uw \in Del(V_{NBR}(u))$ and $\nexists v \in V_{NBR}(u)$ such that $u, w \in V_{NBR}(v)$ and $uw \notin Del(V_{NBR}(v))$. Construction of LDel obviously requires 2-hop neighborhood information. LDel has unbounded degree. In [18], the degree of LDel is limited to $19 + 2\pi/\alpha$, where $0 < \alpha \leq \pi/3$, by applying Yao [19] structure, without sacrificing its spanning property. Note that Yao graph itself does not guarantee planarity.

III. HYPOCOMB

Given as creating points a vertex set V in the Euclidean plane, we show how to build a novel connected planar graph, named *Hypocomb*, by adding edges between them. This is equivalent to removing edges from a complete graph of V . For ease of understanding, we divide our graph planarization process into two steps and present them separately.

For all $a, b \in V$ and $a \neq b$, they are said *collinear* if they have the same X or Y coordinate. Define *north* (*south*) as the positive (resp., negative) direction of the Y axis, and *east* (*west*) as the positive (resp., negative) direction of the X axis. $T = \{\text{north}, \text{west}, \text{south}, \text{east}\}$. For each $dir \in T$, \hat{dir} is the opposite direction, and \check{dir} the set of perpendicular directions. For example, if $dir = \text{north}$, then $\hat{dir} = \text{south}$ and $\check{dir} = \{\text{west}, \text{east}\}$. The border of V is the smallest rectangle containing V and parallel to the two axes.

A. The first step: Besh

At the first step, we build an auxiliary structure, referred to as *Besh* [11]. We synchronously grow from all $v \in V$ four rays $R_v^{\text{north}}, R_v^{\text{west}}, R_v^{\text{south}}$ and R_v^{east} with mutual angle of $\frac{\pi}{2}$, respectively in the north, west, south and east directions. The growth of these rays is limited by the border of V . If there

was no further constraint, we would obtain a mesh. However we indeed apply a blocking rule [11] to control ray growth.

Definition 1 (Blocking rule): $\forall a, b \in V, a \neq b$ and $\forall dir, dir' \in T, dir \neq dir'$, if R_a^{dir} and $R_b^{dir'}$ meet at point u , R_a^{dir} will stop growing only in any of the following cases:

- 1) $|au| > |bu|$;
- 2) $|au| = |bu|, dir' \in \hat{dir}$ and $dir = \text{north}$ or south ;
- 3) $|au| = |bu|$ and $dir' = \check{dir}$.

When this happens, we say ‘ b blocks a at u . In the first two cases (*orthogonal blocking*), it is expressed as $b \overset{u}{\not\leftrightarrow} a$ (or, $R_b^{dir'} \overset{u}{\not\leftrightarrow} R_a^{dir}$); in the last case (*collinear blocking*), it is expressed as $a \overset{u}{\not\leftrightarrow} b$ (or, $R_a^{dir} \overset{u}{\not\leftrightarrow} R_b^{dir'}$).

Use of the blocking rules causes some rays to stop growing early, before hitting the border of V , and yields a truncated mesh, which is our so-called *Besh* (standing for *blocked mesh*). The *Besh*, denoted by $BS(V)$, is defined by a vertex set and an edge set. The former contains the creating points V and added *Besh points*, where the blocking rule is engaged; the later contains the edges between the vertices. In $BS(V)$, each cell is a rectangle. The creating points (*i.e.*, vertices in V) whose rays define the perimeter of a cell is called the *defining points* of the cell. Each cell obviously has at least two, and at most four, defining points. For a Besh cell, with respect to a given corner vertex (which is either a defining point or a Besh point), the *diagonal defining points* are the defining points that are not collinear with the vertex.

Figure 1(a) shows a Besh structure created using 8 points a, b, \dots, h , whose border is marked by a thick rectangle. The solid small dots are Besh points; the thin dashed lines are Besh edges. Besh cell $bswt$ is defined by a, b and h . For this cell, the diagonal defining points with respect to b is a and h , and that with respect to s is a . Examples of blocking case 2 are $b \overset{u}{\not\leftrightarrow} a$, $a \overset{t}{\not\leftrightarrow} b$ and $h \overset{w}{\not\leftrightarrow} a$. An example of blocking case 3 is $b \overset{s}{\not\leftrightarrow} c$. Notice that $|bs| = |cs| < |ds| = |hs|$. By the blocking rule, d (similarly, h) can be blocked by all the other three vertices at s . To reduce this ambiguity, we define over the blocking rule the following important prioritized blocking policy, by which only the blocking from b and c is recognized at s in Fig. 1(a).

Definition 2 (Prioritized blocking): $\forall a, b, c \in V$ and $a \neq b \neq c$, when $a \overset{u}{\not\leftrightarrow} b$ and $c \overset{u}{\not\leftrightarrow} b$ are both possible for the same u , b is considered being blocked by c rather than a .

Definition 3 (Quadrant): Given a point a , $\forall dir \in T$ and $\forall dir' \in \check{dir}$, R_a^{dir} and $R_a^{dir'}$ define a quadrant $Q_a(dir, dir')$. As such, a has four different quadrants.

Lemma 1: $\forall a, b \in V, a \neq b, dir \in T, dir' \in \check{dir}, b \in$

$Q_a(dir, dir')$ and $|bb'| \leq |ab'|$ with b' being the projection of b on R_a^{dir} , if $\exists c \in V$, $c \in Q_a(dir, dir')$ and $c \neq b$ such that c blocks $R_b^{dir'}$ at u , then $|cc'| \leq |ac'|$ where c' is the projection of c on R_a^{dir} .

Proof: We prove this lemma by case study with illustrations being given in Fig. 1(b). According to the way that c blocks $R_b^{dir'}$, we have three cases to consider.

- 1) $R_c^{dir} \not\stackrel{u}{\dashv} R_b^{dir'}$: This is the case of $c = c_1$, $c' = c'_1$ and $u = u_1$. We know $|bb'| \leq |ab'|$ and $|cu| \leq |bu|$. Then $|cc'| = |bb'| - |bu| \leq |ab'| - |cu| = |ac'|$.
- 2) $R_c^{dir} \not\stackrel{u}{\dashv} R_b^{dir'}$: This is the case of $c = c_2$, $c' = c'_2$ and $u = u_2$. We have $|cc'| = |bb'| - |bu| \leq |bb'| \leq |ab'| \leq |ab'| + |b'c'| = |ac'|$.
- 3) $R_c^{dir} \stackrel{u}{\dashv} R_b^{dir'}$: This is the case of $c = c_3$, $c' = c'_3 = b'$ and $u = u_3$. $|cc'| < |b'c'| < |ac'|$.

Note that c' is within distance $|bb'|$ from b' . \blacksquare

Lemma 1 tells us an important property of the blocking rule: if a node b blocks a orthogonally at u in the case that a and b are the only vertices in V , then a must be blocked by a vertex c (possibly identical to b) orthogonally at u' within distance $|bu|$ from u when V contains also other vertices. On the basis of this result, we develop a computer algorithm named *Blocking-Detection* to support Besh construction. Given $a \in V$ and $dir \in T$, this algorithm returns the set of vertices (at most 2 by the prioritized blocking policy) that block R_a^{dir} and the associated Besh point (a single point). If no vertex blocks R_a^{dir} , it returns an empty set. The pseudo codes are given in Algorithm 1. Functions *First()* and *Second()* return respectively the first and the second element of an input pair. Function *arg()* returns the argument of an input function.

Examine Algorithm 1. In Line 3, we find the vertex c that is located on R_a^{dir} and nearest to a in $O(|V|)$ time. Assume that R_a^{dir} is not blocked by anybody else. This vertex c will collinearly block R_a^{dir} at the mid point d of a and c if R_c^{dir} is not orthogonally blocked before reaching d . Thus we perform further check on its this blocking potential. In Lines 7 and 8 we compute the sets S_1 and S_2 of vertices (together with the corresponding blocking points) that have the potential to orthogonally block R_c^{dir} . The computation can be finished in $O(|V|)$ time. In the light of Lemma 1, we in Line 9 reduce S_1 to S'_1 by removing the vertices that are not able to block R_c^{dir} before d , due to being blocked by other vertices in S_1 . The computation time is at most $O(|V|^2)$. Lines 10 reduces S_2 to S'_2 in a similar way in $O(|V|^2)$ time. If the union of S'_1 and S'_2 is empty (namely, no vertex blocks R_c^{dir} orthogonally), then we can conclude that R_c^{dir} is able to block R_a^{dir} . Otherwise, R_c^{dir} will not block R_a^{dir} , and R_a^{dir} will reach the point where R_c^{dir} is orthogonally blocked by a vertex and be blocked by that same vertex. Hence, the result from Lines 4-13 is a coarse upper bound of the length of R_a^{dir} in Besh, stored in variable $dist$. It is infinity in the case that c does not exist (without considering the constraint from the border of V).

The upper bound $dist$ is derived under the assumption that R_a^{dir} is not blocked by anybody else. In the latter half of the algorithm, we remove this assumption. Lines 15 and 16

Algorithm 1 Blocking-Detection(V, a, dir)

```

Require:  $a \in V$  and  $dir \in T$ 
1:  $dist := \infty$ 
2: Let  $dir'$  and  $dir''$  be the two elements in  $\hat{dir}$ 
3:  $c := \arg(\min_{b \in V, b \neq a, b \in R_a^{dir}} |ab|)$ 
4: if  $c \neq null$  then
5:    $dist := \frac{1}{2}|ac|$ 
6:    $d :=$  mid point of  $a$  and  $c$ 
7:    $S_1 := \{(b, p) | b \in V \text{ such that } R_b^{dir'} \dashv R_c^{dir} \text{ in the case of } V = \{b, c\}\}$ 
8:    $S_2 := \{(b, p) | b \in V \text{ such that } R_b^{dir''} \dashv R_c^{dir} \text{ in the case of } V = \{b, c\}\}$ 
9:    $S'_1 := \{m | m \in S_1 \text{ and } |Second(m)c| \leq dist \text{ such that } \nexists t \in S_1, \\ \text{First}(t) \text{ blocks } R_{First(m)}^{dir'} \text{ in the case of } V = \{First(m), First(t)\}\}$ 
10:   $S'_2 := \{m | m \in S_2 \text{ and } |Second(m)c| \leq dist \text{ such that } \nexists t \in S_2, \\ \text{First}(t) \text{ blocks } R_{First(m)}^{dir''} \text{ in the case of } V = \{First(m), First(t)\}\}$ 
11:  if  $S'_1 \cup S'_2 \neq \emptyset$  then
12:     $dist := |Second(\arg(\min_{m \in S'_1 \cup S'_2} |Second(m)c|))a|$ 
13:  end if
14: end if
15:  $W_1 := \{(b, p) | b \in V \text{ such that } R_b^{dir'} \dashv R_a^{dir} \text{ in the case of } V = \{a, b\}\}$ 
16:  $W_2 := \{(b, p) | b \in V \text{ such that } R_b^{dir''} \dashv R_a^{dir} \text{ in the case of } V = \{a, b\}\}$ 
17:  $W'_1 := \{m | m \in W_1 \text{ and } |Second(m)a| \leq dist \text{ such that } \nexists t \in W_1, \\ \text{First}(t) \text{ blocks } R_{First(m)}^{dir'} \text{ in the case of } V = \{First(m), First(t)\}\}$ 
18:  $W'_2 := \{m | m \in W_2 \text{ and } |Second(m)a| \leq dist \text{ such that } \nexists t \in W_2, \\ \text{First}(t) \text{ blocks } R_{First(m)}^{dir''} \text{ in the case of } V = \{First(m), First(t)\}\}$ 
19: if  $W'_1 \cup W'_2 \neq \emptyset$  then
20:    $dist' := \min_{m \in W'_1 \cup W'_2} |Second(m)a|$ 
21:    $ret := \{m | m \in W'_1 \cup W'_2 \text{ such that } |Second(m)a| = dist'\}$ 
22: else if  $c \neq null$  then
23:    $ret := \{(c, d)\}$ 
24: else
25:    $ret := \emptyset$ 
26: end if
27: return  $ret$ 

```

compute the sets W_1 and W_2 of vertices that have the potential to block R_a^{dir} in $O(|V|)$ time; Line 17 and 18 reduce W_1 and W_2 to W'_1 and W'_2 respectively, by removing the vertices that are not able to orthogonally block R_a^{dir} in $O(|V|^2)$ time. The computation in these four lines is similar to that in Lines 7-10. In the case that the union of W'_1 and W'_2 is not empty, the associated blocking point nearest to a in these two sets is identified in $O(|V|)$ time (Line 20). It is the true blocking point, *i.e.*, Besh point. The rationale is that R_a^{dir} can be blocked only at a single point, and after that no blocking is possible at any point further away from a . Thus the set of blocking vertices associated with this point are found and returned (Line 21). This final step takes another $O(|V|)$ time. In the case that the union of W'_1 and W'_2 is empty, the return value is single-element set $\{(c, d)\}$ if c exists (Line 23), and \emptyset otherwise (Line 25). The computation time is constant $O(1)$. All the other lines in the algorithm take $O(1)$ time.

In total $O(|V|^2)$ is the complexity of Algorithm 1. More efficient algorithms may be developed, but beyond the scope of this paper. The correctness of Algorithm 1 simply follows from the above analysis. Then we may construct Besh within $O(|V|^3)$ time, by running this algorithm for every vertex in V four times, each time for a different direction in T .

Although Besh is a transit product of our graph planarization process, it has its own importance in real life applications. In [11], we derived that Besh has good proximity property like Voronoi diagram through analytical study and simulation experiments, showed how to accomplish Besh in a localized way, without knowing V , and proposed a Besh-based localized

distance-sensitive service discovery algorithm for wireless sensor and actor networks.

Before proceeding to the second drawing step, we would like to introduce a few important definitions and lemmas (whose proof can be found in [10]) to be used in the sequel.

Definition 4 (Emptiness and Cleanness): A region is *empty* if and only if there are no vertices located in it; a region is *clean* (with respect to Besh) if and only if it does not contain any Besh edge. A clean region must be empty, while the converse is obviously not necessarily true.

Lemma 2: $\forall a, b \in V, a \neq b, dir \in T$ and $dir' \in \hat{dir}$, if $R_a^{dir} \not\leftrightarrow R_b^{dir'}$ then the region defined by triangle Δab (including its perimeter) is empty.

Lemma 3: $\forall a, b \in V$ and $a \neq b$, if they have a mutual blocking relation, then $\square ab$ is clean in $BS(V)$. Here $\square ab$ is the region defined by the rectangle (including its perimeter) parallel to the X and Y axes and with ab being diagonal line.

Lemma 4: $\forall a, b \in V$ and $R_a^{dir} \not\leftrightarrow R_b^{dir'}$ in $BS(V)$, if $R_b^{dir'} \leftrightarrow R_a^{dir}$ is not present in $BS(V)$ (namely, there is no mutual blocking between a and b), then

- 1) for the Besh cell $BC_w(\overline{dir}, \overline{dir'})$ cornered at w in $Q_w(\overline{dir}, \overline{dir'})$, there is exactly one diagonal definition point c with respect to w , and
- 2) c has a blocking relation with both a and b , and
- 3) $\max(mlen(\square ac), mlen(\square bc)) < mlen(\square ab)$, where $mlen(\cdot)$ is the length of the longest side of the box.

B. The second step: ‘Dual’ of Besh

Having obtained $BS(V)$, we start the second step. At this step, we create the ‘dual’ of $BS(V)$ by adding edges between the creating points (points in V) that have a blocking relation. Here term ‘dual’ is from an abused use of duality. It is of importance to remember that inter-vertex blocking relation is subject to the prioritized blocking policy. Formally, we define

Definition 5 (HC edge creation rule): $\forall a, b \in V$ and $a \neq b$, create edge ab if and only if a and b have a blocking relation.

The dual of Besh $BS(V)$ is composed of the given vertex set V and the added edge set. We name it *Hypocomb* (standing for *Hypotenuse-comb*) and denote it by $HC(V)$. The name ‘Hypocomb’ owns its inspiration to the fact that each edge ab due to $a \not\leftrightarrow b$ is the hypotenuse of the right triangle Δab . In Fig. 1(a), Hypocomb edges are drawn in thick links. By the HC edge creation rule and Algorithm 1, we can trivially build Hypocomb in $O(|V|^3)$ time. Below we analyze the connectivity, planarity and degree bound of Hypocomb.

Theorem 1: $HC(V)$ is connected.

Proof: For all $a, b \in V$ and $dir, dir' \in T$, if $R_b^{dir'} \leftrightarrow R_a^{dir}$ or $R_b^{dir'} \leftrightarrow R_a^{dir}$, then we say $R_b^{dir'}$ is an extension of R_a^{dir} . A ray has at most 2 extensions. In Fig. 1(a), R_d^{south} is extended by both R_c^{south} and R_b^{south} , for example. Ray extension occurs from a toward dir in a cascaded fashion until a vertex, called *terminal node*, whose ray growing in direction dir is not blocked (by any other vertex) is reached. Cascaded ray extension defines a directed acyclic graph $DAG(a, dir)$, where nodes are the vertices involved and edges imply direct ray extension relation. Define $DAG(a, dir, \overline{dir}) = DAG(a, dir) \cup$

$DAG(a, \overline{dir})$. It spans the space enclosed by the border of V . Because direct ray extension implies blocking relation, each edge in $DAG(a, dir, \overline{dir})$ corresponds to an edge with the same end nodes in $HC(V)$. As such, this DAG is mapped to a subgraph of $HC(V)$, denoted by $M_{DAG}(a, dir, \overline{dir})$, which is connected due to the reachability from a to every other node in $DAG(a, dir, \overline{dir})$. For all $a' \in V, a' \neq a$ and $dir' \in \hat{dir}, DAG(a', dir', \overline{dir'})$ must have some node(s) in common with $DAG(a, dir, \overline{dir})$. It is due to the spanning property and perpendicularity of the two DAGs. As a consequence, $M_{DAG}(a, dir, \overline{dir})$ and $M_{DAG}(a', dir', \overline{dir'})$ are connected. By definition, $HC(V) = \bigcup_{a \in V} (M_{DAG}(a, dir, \overline{dir}) \cup M_{DAG}(a, dir', \overline{dir'}))$. The connectivity of $HC(V)$ follows. ■

Theorem 2: $HC(V)$ is planar.

Proof: Assume for the sake of contradiction $ab, cd \in HC(V)$ and they intersect. Let u be a blocking point of a and b , and let v be a blocking point of c and d . Consider the two triangles Δaub and Δcvd . By Lemma 2, they are both empty, that is to say, $a, b \notin \Delta cvd$ and $c, d \notin \Delta aub$. Then the two triangles must intersect, with their hypotenuses being across. In this case, one of the catheti of Δaub , say au , intersects with one of the catheti, say cv , of Δcvd . Let the crossover point be w . A blocking relation between a and c occurs at w . This renders either the blocking of a and b at u or that of c and d at v invalid. A contradiction is reached. ■

Theorem 3: $\Delta(HC(V)) \leq |V| - 1$.

Proof: It is obvious that $\Delta(HC(V))$ can not be larger than $|V| - 1$ which is the degree of the complete graph of V . We just need to show that it is possible to have $\Delta(HC(V)) = |V| - 1$. Examine a particular vertex arrangement given in Fig. 1(c), where $|au| = |cu|$. Any vertex on the line segment bc will be blocked by a , and thus has an incidental edge with a in the corresponding Hypocomb. If all the other vertices in V are located on bc , vertex a will have degree exactly $n - 1$. This completes the proof. ■

IV. REDUCED HYPOCOMB

In previous section we presented a novel planar graph, Hypocomb, which is extracted from a complete graph and has unbounded degree. In this section we simplify Hypocomb, reducing the number of edges, by applying a constrained edge creation rule (see Definition 6) at the second drawing step. We refer to the resultant simplified Hypocomb as *Reduced Hypocomb* and denote it by $RHC(V)$.

Definition 6 (RHC edge creation rule): $\forall a, b \in V$ and $a \neq b$, create edge ab iff a and b have a *mutual* blocking relation.

Corollary 1: $RHC(V) \subseteq HC(V)$.

In Fig. 1(a), only solid thick lines belong to Reduced Hypocomb. Corollary 1 is derived immediately from the RHC edge creation rule. With Algorithm 1, Reduced Hypocomb construction is straightforward and has the same complexity $O(|V|^3)$ as Hypocomb construction. In the following we show that Reduced Hypocomb not only remains connected planar but also possesses the desired bounded-degree property.

Theorem 4: $RHC(V)$ is connected.

Proof: Since $RHC(V)$ is a subgraph of $HC(V)$, the construction of $RHC(V)$ can be viewed an edge removal

process in $HC(V)$, where we remove non-RHC edges one by one. Consider an arbitrary non-RHC edge $ab \in HC(V)$. By definition, a and b have no mutual blocking relation. Without loss of generality, let the inclusion of ab in $HC(V)$ is due to $R_a^{dir} \xrightarrow{w} R_b^{dir'}$ with $dir \in T$ and $dir' \in \bar{dir}$. By Lemma 4, we have $ac, bc \in HC(V)$ where c is the unique diagonal definition point of the Besh cell cornered at w and located in $Q_w(\bar{dir}, \bar{dir}')$ with respect to w . If we remove ab and only ab from $HC(V)$, a and b remain connected via c . We call such an edge removal action ‘connectivity division’ and call ac and bc the results of division of ab by c . Because it is possible that ac and bc are also removed, connectivity division would not preserve connectivity unless no division loop is induced.

Below we prove that no division loop occurs. Assume for the sake of contradiction that there are division loops. Take a smallest loop where each edge appears only once. We express this loop by $u_0v_0 \xrightarrow{w_1} u_1v_1 \xrightarrow{w_2} u_2v_2 \xrightarrow{w_3} \dots \xrightarrow{w_n} u_nv_n$. Let $u_nv_n = u_0v_0$. For $i = 1, \dots, n$, $u_{i-1}v_{i-1} \xrightarrow{w_i} u_iv_i$ indicates that u_iv_i is a result of the connectivity division of $u_{i-1}v_{i-1}$ by w_i , where $u_i \in \{u_{i-1}, v_{i-1}\}$ and $v_i = w_i$. We know u_{i-1} and v_{i-1} have no mutual blocking relation and $R_{u_{i-1}}^{dir_{i-1}} \xrightarrow{w_i} R_{v_{i-1}}^{dir'_{i-1}}$ (or $R_{v_{i-1}}^{dir'_{i-1}} \xrightarrow{w_i} R_{u_{i-1}}^{dir_{i-1}}$) for $dir_{i-1} \in T$ and $dir'_{i-1} \in \bar{dir}_{i-1}$. Recall that $mlen(\square u_{i-1}v_{i-1})$ is the maximum side length of $\square u_{i-1}v_{i-1}$. In this case, applying lemma 4 along the division loop, we have $mlen(\square u_0v_0) > mlen(\square u_1v_1) > \dots > mlen(\square u_{n-1}v_{n-1}) > mlen(\square u_0v_0)$, which is impossible. ■

Theorem 5: $RHC(V)$ is planar.

Proof: It follows from Theorem 2 and Corollary 1. ■

Lemma 5: $\forall ab, ac \in RHC(V)$, creation of ab is due to $R_a^{dir} \xrightarrow{u} R_b^{dir'}$ and ac is due to $R_a^{dir'} \xrightarrow{w} R_c^{dir''}$ with $dir \in T$ and $dir' \in \bar{dir}$, $\nexists ad \in RHC(V)$ such that $ad \in Q_a(dir, dir')$.

Proof: Assume for the sake of contradiction that such ad exists. By definition, a and d must have a mutual blocking relation. Without loss of generality, let $R_a^{dir} \xrightarrow{u} R_d^{dir'}$ and symmetrically $R_d^{dir'} \xrightarrow{w} R_a^{dir''}$ for some u and w . However $R_d^{dir'} \xrightarrow{w} R_a^{dir''}$ and $R_a^{dir'} \xrightarrow{u} R_c^{dir''}$ can not hold at the same time (even if $w = t$, by the prioritized blocking policy). ■

Theorem 6: $\Delta(RHC(V)) \leq 6$.

Proof: $\forall a \in V$, there are at most 4 clean $\square ab$ in $BS(V)$ in the four quadrants of a (one in each quadrant), and at most 4 clean $\square ab$ (which reduces to ab) along the X and the Y axis respectively in the four directions. Hence a has at most 8 incidental edges in $RHC(V)$, 4 quadrant edges and 4 axis edges. By Lemma 5, two axis edges must be either separated by more than one quadrant edge or adjacent to each other. This constraint then lowers the upper bound to 6. The scenario of degree 6 is that a has 4 quadrant edges and 2 collinear axis edges either along the X axis or the Y axis. ■

V. LOCAL HYPOCOMB

Till now, we have successfully bounded the degree of Hypocomb above by a small constant 6, by applying constrained edge creation rule and yet without jeopardizing its connectivity and planarity properties. Hypocomb and Reduced Hypocomb are built with complete knowledge of V and with no constraint on edge length, *i.e.*, extracted from a complete

graph of V . In this section we investigate how to build Reduced Hypocomb on UDG with limited local knowledge. UDG has the following important property (proof is in [10]).

Lemma 6: In UDG, if two edges intersect, then one end vertex of one edge neighbors the two end vertices of the other.

Specifically, $\forall a \in V$, when we draw incidental edges for it, we merely have the position information of vertices b located in the unit disk $D_\gamma(a)$ of a . In this case, we propose a local edge creation rule (see Definition 7), which adds ab according to its inclusion in the Reduced Hypocomb graph of the *closed common neighbor set* of a and b . And obviously, the creation decision on edge ab is symmetric for a and b .

Definition 7 (LHC edge creation rule): $\forall a \in V, b \in V_{NBR}(a)$ and $a \neq b$, create edge ab if and only if $ab \in RHC(V_{NBR}(a, b))$.

This local edge creation rule is dependent on 1-hop neighbors position information only. It gives the graph construction process strictly localized feature. The resultant Hypocomb variant is therefore called *Local Hypocomb*, and denoted by $LHC(V)$. We know $|V_{NBR}(a, b)| \leq d(a) + 1$, where $d(a)$ is the degree of a in UDG. Using Algorithm 1 each node a is able to build $RHC(V_{NBR}(a, b))$ for each neighbor b in $O(d(a)^3)$ time, and the total cost (per node) of determining LHC edges is therefore $O(d(a)^4)$. Because $d(a) \leq \Delta(UDG(V))$, an upper bound of the computation cost on each node for Local Hypocomb construction is $O(\Delta(UDG(V))^4)$. Below we show that Local Hypocomb surprisingly remains connected planar and has degree bounded above by 8 (just slightly larger than the degree bound 6 of Reduced Hypocomb).

Theorem 7: $LHC(V)$ is connected.

Proof: We view $LHC(V)$ construction as an edge removal process in $UDG(V)$. For every $ab \in UDG(V)$ and $ab \notin LHC(V)$ (*i.e.*, a removed edge), by definition we have $ab \notin RHC(V_{NBR}(a, b))$. This implies either $ab \notin HC(V_{NBR}(a, b))$ or, $ab \in HC(V_{NBR}(a, b))$ and there exists a unique $c \in V_{NBR}(a, b)$ such that it divides ab into $ac, bc \in HC(V_{NBR}(a, b))$ (Lemma 4). In the former case, the removal of ab does not affect the connectivity between a and b since we know $HC(V_{NBR}(a, b))$ is connected (Theorem 1). In the later case, a and b remain connected (through c) from the local view of a and b after removing ab . To prove the connectivity of $LHC(V)$, it is sufficient to prove that local connectivity division (edge removal) actions do not generate division loop in a global sense. The loop-free property can be proved similarly as in Theorem 4. The key is to explore the stability of $mlen(\square ab)$ (*i.e.*, it is the same in any vertex’s local view) and the monotonically decreasing nature of $mlen(\square ac)$ and $mlen(\square bc)$ relative to $mlen(\square ab)$. ■

Lemma 7: Any two crossover edges $ab, cd \in UDG(V)$ do not belong to $LHC(V)$ at the same time if $ac, ad \in UDG(V)$ and one of bc and bd appears in $UDG(V)$.

Proof: Assume for the sake of contradiction $ab, cd \in LHC(V)$. Without loss of generality, let $bc \in UDG(V)$. By definition 7, $ab \in RHC(V_{NBR}(a, b))$. From Lemma 3, $\square ab$ is clean in $BS(V_{NBR}(a, b))$ and thus $c \notin \square ab$; likewise, $\square cd$ is clean in $BS(V_{NBR}(c, d))$ and $a \notin \square cd$. Under these

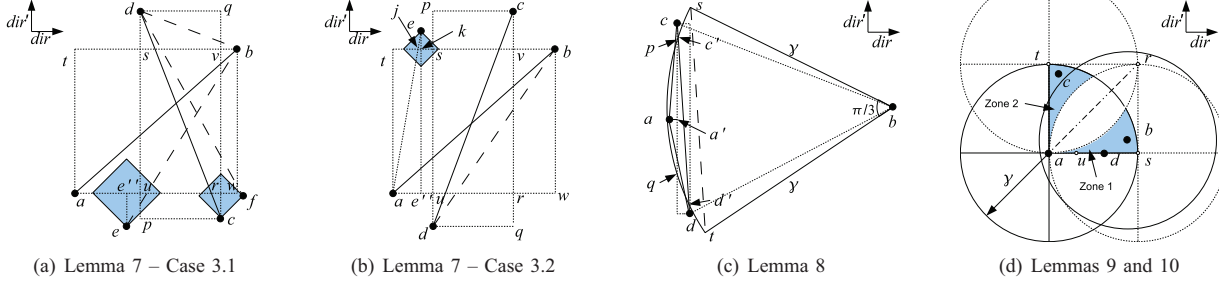


Fig. 2. Illustrations for Local Hypocomb

constraints, by varying the relative position of b and d to $\square cd$ and $\square ab$ we obtain the following cases: (1) $b \in \square cd$ and $d \notin \square ab$; (2) $d \in \square ab$ and $b \notin \square cd$; (3) $b \notin \square cd$ and $d \notin \square ab$. Obviously, $b \in \square cd$ and $d \in \square ab$ can not hold at the same time; thus this case is not in our consideration. In case (1), $|bd| \leq |cd|$ because $\angle cbd$ is not acute in triangle Δcbd . In case (2), $|bd| \leq |ab|$ because $\angle adb$ is not acute in Δadb . In these two cases, $d \in V_{NBR}(a, b)$ and $b \in V_{NBR}(c, d)$, and obviously ab and cd do not appear in $LHC(V)$ at the same time, contradicting our assumption. Case (3) has two sub-cases. Below we derive a contradiction from both of them.

Case 3.1 (Fig. 2(a)): We first put ourselves under the condition: a would block d at u if V contained only a, d . We have $|au| \leq |du|$ by the blocking rule. Because $a \in V_{NBR}(c, d)$ and $cd \in RHC(V_{NBR}(c, d))$, R_a^{dir} must be blocked by a vertex e_0 in $BS(V_{NBR}(c, d))$ before reaching segment dp . If e_0 has a projection e'_0 on dp , then by Lemma 1 we have $|e_0e'_0| \leq |de'_0|$ and are facing the same situation as with a , and therefore the same argument can be made for e_0 . By these means, we are presented a blocking chain in $BS(V_{NBR}(c, d))$ that ends at a vertex e_n ($n \geq 0$) that has no projection on dp . Let e''_i be the projection of e_i on R_a^{dir} . By the blocking rule and Lemma 1, we easily have $|e_i e''_i| \leq |ae''_i| < |au| < |du|$ for $i = 0, \dots, n$. Since $|du| > |au|$, e_n can not be around vertex d but point p . For ease of presentation, let $e = e_n$ and $e'' = e''_n$, as shown in the figure. Notice $|cr| = |pu| \leq |ee''| \leq |ae''| < |au| < |ar|$. This implies that c is in the same situation with respect to a as a with respect to d . By the same argument, we conclude that there exists such a vertex f around w for c (like e for a).

By simple geometry, the four vertices a, b, c and d are all neighboring e if $|su| \leq |sv|$, and f otherwise. Without loss of generality, we consider $|su| \leq |sv|$ since the other case is symmetric. In $BS(V_{NBR}(a, b))$, R_e^{dir} must be blocked by a vertex g_0 at a point x on segment ee'' . According to the blocking rule, g_0 must be located in a square area (shaded in the figure) with e as corner and with the diagonal line defined by two other corners lying on segment aw . Trivially, all points in the square are common neighbor of a, b, c and d . Since in $BS(V_{NBR}(c, d))$, g_0 does not block R_e^{dir} , there must be a vertex g_1 that blocks g_0 at a point y on segment g_0x . This vertex g_1 is again located in the square area and thus neighboring a, b, c and d . The argument can be made iteratively, alternate between $BS(V_{NBR}(a, b))$ and $BS(V_{NBR}(c, d))$, giving us a set of vertices g_0, \dots, g_m all located in the square area and neighboring a, b, c and d . Hence the blocking relations among

them appear in both $BS(V_{NBR}(a, b))$ and $BS(V_{NBR}(c, d))$, and ab and cd can not appear in $LHC(V)$ at the same time, contradicting our assumption.

We now consider the opposite condition: d would block a if a and b were the only vertices in V . We have $|au| \geq |du|$ ($\geq |ds|$) and $|bd| < |ds| + |bs| \leq |au| + |bs| = |bt| \leq |ab|$, i.e., $d \in V_{NBR}(a, b)$. Then the only situation worth investigation includes the combination of the following conditions: b would block d at s if V contained only b, d ($|ds| \geq |bs| \geq |bv|$), c would block b at v if V contained only b, c ($|bv| \geq |cv| \geq |cr|$), and a would block c at r if V contained only a, c ($|cr| \geq |ar| \geq |au|$). It is because any other situation is equivalent to the previous one after vertex renaming and therefore leads to a similar contradiction. Under this circumstance, we have $|au| = |ds| = |bv| = |cr| = |au|$ and $|su| = |sv| = 0$. Then either a and b or, c and d , do not have blocking relation according to the prioritized blocking policy, a contradiction.

Case 3.2 (Fig. 2(b)): As in case (3.1), we first investigate under the condition that a would block d at u if V contained only a, d . Then with respect to a , we may conclude a similar set of vertices $e_0, \dots, e_n \in V_{NBR}(c, d)$ in order to enable $cd \in RHC(V_{NBR}(c, d))$. Among them, e_n is around p . Let e be the one closest to segment bt and on the same side as p and e'' the projection of e on R_a^{dir} . In right triangle $\Delta ee''a$, $|ae|^2 = |ae''|^2 + |ee''|^2 < |au|^2 + |du|^2 = |ad|^2$. Let j be the intersection point of ae and bt and k the intersection point of ee'' and bt . Trivially, $|ek| \leq |kj|$. In right triangle Δekb , $|be|^2 = |ek|^2 + |kb|^2 \leq |kj|^2 + |kb|^2 \leq (|tj| + |kj| + |kb|)^2 + |at|^2 = |ab|^2$. Hence $e \in V_{NBR}(a, b)$. By the same technique we may derive a similar contradiction as in case (3.1).

Likewise, we can derive a contradiction under the condition that c would block b at v if V contained only b, c . We only remain to consider the combination of the opposites of the two conditions, where $|au| \geq |du|$ and $|cv| \geq |bv|$. Observe $|bt| = |au| + |sv| + |bv|$, $|dp| = |cv| + |at| + |du|$, $|ds| = |at| + |du|$ and $|cp| = |sv|$. Let $\delta_1 = |bd|^2 - |ab|^2$ and $\delta_2 = |bd|^2 - |cd|^2$. In right triangle Δcpd , $|cd|^2 = |dp|^2 + |cp|^2 = (|cv| + |at| + |du|)^2 + |sv|^2$. In right triangle Δbsd , $|bd|^2 = |bs|^2 + |ds|^2 = (|bv| + |sv|)^2 + (|at| + |du|)^2$. In right triangle Δatb , $|ab|^2 = |at|^2 + |bt|^2 = |at|^2 + (|au| + |sv| + |bv|)^2$. Then $\delta_1 = |du|^2 - |au|^2 + 2|du||at| - 2|au|(|sv| + |bv|)$ and $\delta_2 = |bv|^2 - |cv|^2 + 2|bv||sv| - 2|cv|(|at| + |du|)$. Recall $|du| \leq |au|$ and $|bv| \leq |cv|$. If $|at| \leq |sv| + |bv|$, then $\delta_1 \leq 0$ (i.e., $|bd| \leq |ab|$); otherwise, $|sv| < |at| - |bv| < |at| + |du|$ and thus $\delta_2 \leq 0$ (i.e., $|bd| \leq |cd|$). This implies $bd \in UDG(V)$. Thus current

situation is equivalent to the first situation examined (after switching the name of a and d and other vertex remaining), and we may derive a contradiction similarly. ■

Lemma 8: Any two crossover edges $ab, cd \in UDG(V)$ do not belong to $LHC(V)$ at the same time if $ac, ad \in UDG(V)$ and $bc, bd \notin UDG(V)$.

Proof: Clearly, cd must intersect the unit circle $C_\gamma(b)$ of b as, otherwise, $ab \notin UDG(V)$. Let c' and d' be intersection points of cd and $C_\gamma(b)$. $|cd| > |c'd'|$. Then c' and d' must be on arc st of $\pi/3$ of $C_\gamma(b)$, with chord st parallel to cd , as shown in Fig. 2(c). It is because, otherwise, $|cd| > \gamma$ (given cd intersects ab) can not belong to $UDG(V)$. In this case and being with the constraint $ab \in UDG(V)$, a must be located in the arc segment area defined by c' and d' . And, it must be located outside $\square cd$ so that $cd \in RHC(V_{NBR}(c, d))$. This additional restriction limits the location of a to be within the arc segment defined by the intersection points p and q of $\square cd$ and arc $c'd'$. In Fig. 2(c), $\square cd$ is shown by a dotted rectangle. The tangent of $C_\gamma(b)$ at s has a $\pi/6$ angle with st . The angle of the tangent at s therefore has an angle less than $\pi/6$ with pq . We have $\angle acq < \angle apq < \pi/6 < \pi/4$. Recall $a \in V_{NBR}(c, d)$. In right triangle $\triangle ca'a$, where a' is the projection of a on cq , $|aa'| < |ca'|$. This implies $R_a^{dir} \xrightarrow{a'} R_c^{dir'}$ if no other vertex blocks R_a^{dir} before it reaches $R_c^{dir'}$. It is possible some vertex $m \in V_{NBR}(c, d)$ blocks a such that this blocking relation does not exist. However, in this case, m will block $R_c^{dir'}$ if no other vertex blocks m by Lemma 1. The same argument can be made iteratively. Since we have a finite number of vertices in $V_{NBR}(c, d)$, finally a vertex will block $R_c^{dir'}$. And obviously this vertex must be located in either of the two squares with aa' as common edge. These two squares are between p and q due to the fact that $\angle apq < \pi/4$ and $\angle pqa < \pi/4$. Thus the mutual blocking relation of c and d is broken. This finally contradicts $cd \in RHC(V_{NBR}(c, d))$. ■

Theorem 8: $LHC(V)$ is planar.

Proof: Any edge in $LHC(V)$ is also in $UDG(V)$. For any pair of crossover edges ab and cd in $UDG(V)$, without loss of generality, let $ac, ad \in UDG(V)$ by Lemma 6. Then regardless the containment relations of bc and bd in $UDG(V)$, ab and cd do not appear in $LHC(V)$ at the same time according to Lemma 7 and 8. Thus the theorem holds. ■

Lemma 9: $\forall dir \in T, dir' \in \hat{dir}, ab, ac \in LHC(V)$, $ab, ac \in Q_a(dir, dir')$ and $ab \neq ac$, $\nexists ad \in LHC(V)$ and $ad \neq ab, ac$ such that $ad \in Q_a(dir, dir')$.

Proof: We first derive $|bc| > \gamma$, where γ is unit distance. It is because, otherwise, b and c would be in each other's closed common neighborhood with a , and in this case, by Lemma 5 at most one of ab and ac would belong to $LHC(V)$. Then c must be located in the differential area of $D_\gamma(a)$ and the $D_\gamma(b)$ in $Q(dir, dir')$. Let s and t respectively be the intersection point of R_a^{dir} and $R_a^{dir'}$ with $C_\gamma(a)$, as shown in Fig. 2(d). For such a residence area of c to exist, b must be in one of the shaded areas, which are defined by $C_\gamma(s)$ and $C_\gamma(t)$. Symmetrically, c must be in the other shaded area. For the sake of contradiction, assume $\exists ad \in LHC(V)$, $ad \neq ab, ac$,

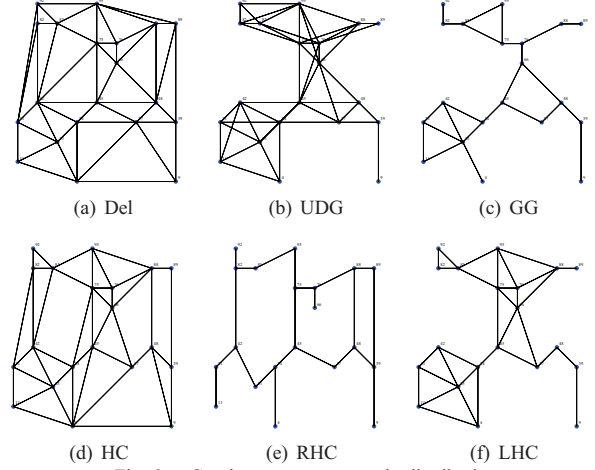


Fig. 3. Graphs over a same node distribution

and $ad \in Q_a(dir, dir')$. Then $b \notin D_\gamma(d)$ and $c \notin D_\gamma(d)$. That is, vertex d must be located in the intersection area of the two shaded areas, which however does not exist. ■

Lemma 10: $\forall ad \in LHC(V)$, if the creation of ad is due to $R_a^{dir} \xrightarrow{u} R_d^{dir'}$ with $dir \in T$, then $\nexists ab, ac \in LHC(V)$ and $ab \neq ac$ such that $\exists dir' \in \hat{dir}, ab, ac \in Q_a(dir, dir')$.

Proof: Assume for the sake of contradiction that such ab and ac exist. Observe Fig. 2(d), which depicts $Q_a(dir, dir')$ in a generic way. Vertices b and c must be located separately in the two shaded areas, as we analyzed in the proof of Lemma 9. Without loss of generality, let b be in zone 1 and c in zone 2. Obviously, $\angle uab < \pi/4$. To ensure the blocking relation $R_a^{dir} \xrightarrow{u} R_d^{dir'}$, we must have $R_a^{dir} \xrightarrow{u'} R_b^{dir'}$ with $u' \in au$ (not shown in the figure). This implies $|bu'| \geq |au'|$ and thus $\angle uab \geq \pi/4$. A contradiction is reached. ■

Theorem 9: $\Delta(LHC(V)) \leq 8$

Proof: It follows immediately from Lemmas 9 and 10. Lemma 9 indicates that in LHC each node has at most 2 edges in each quadrant; Lemma 10 indicates that in LHC, if a node has an axis edge, then it has at most 1 edge in each of the 2 quadrants adjacent to that edge. Thus the scenario of degree 8 is that a vertex has 2 edges in each quadrant or that it has 4 axis edges and 4 quadrant edges, one in each quadrant. ■

VI. NUMERICAL RESULTS

We now study the average degree of Hypocomb family graphs and their impact on FACE routing [2], in comparison with Del and GG, through extensive simulation. We run simulation experiments using a custom C simulator to build these graphs over the same random node (*i.e.*, vertices) distribution. To do so, we compute a virtual $l \times l$ grid and place n nodes at n randomly selected unique grid points. For GG and LHC computation, a UDG is generated with a properly selected unit distance to ensure connectivity. An example construction of these graphs when $n = 20$ and $l = 10$ can be found in Fig. 3. We run FACE over each graph for a randomly picked pair of source and destination. Indeed, FACE has to be run on a planar graph only, and it was supported by GG in [2]. Below we report our numerical results, which are obtained from 1000 simulation runs with $l = 20$ and n varying from 20 to 300.

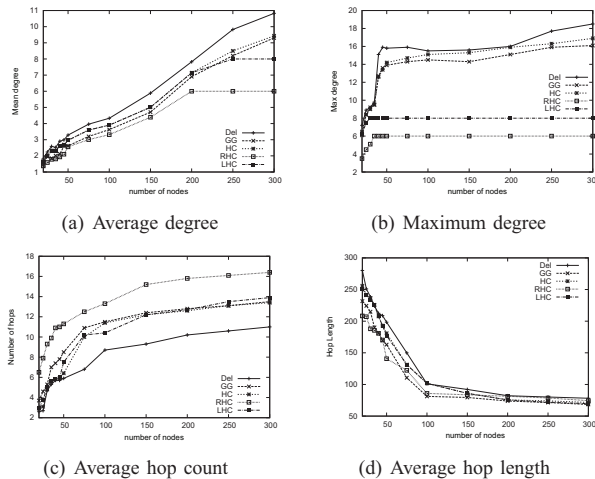


Fig. 4. Numerical results

Figure 4(b) verifies our theoretical findings about degree bound: HC has unbounded degree while the degree of RHC and LHC is bounded above by 6 and 8, respectively. Figure 4(a) shows the average degree (reflecting how sparse or dense a graph is topologically), which as expected slowly increases with the overall number n of nodes. For RHC and LHC, it never exceeds the corresponding degree bound. We observe that their curves become flat after a turning point of $n = 200, 250$ respectively. Del, HC and RHC are extracted from complete graph and therefore comparable to each other. Among them Del and RHC are respectively the densest and the sparsest. GG and LHC are both local graphs and thus competitors. LHC is a bit denser than GG before the turning point ($n = 250$) and is increasingly sparser afterwards as GG has no degree bound. Generally speaking, the higher the average degree (*i.e.*, the denser the network), the smaller average face size, and therefore more likely to find direct paths (composed of relatively long links though). This expectation is confirmed by our simulation results plotted in Fig. 4(c) and 4(d). Notice that for a dense UDG ($n > 250$), although GG is denser than LHC, they lead to almost the same FACE routing performance.

It is well-known that GG contains short edges and FACE routing suffers from long routing paths in a sparse UDG when GG is used for planarization. Our simulation reveals that FACE will benefit from replacing GG with LHC. In addition, note that using the long edges provided by LHC may help in saving energy when used in a ETE [3] fashion, *i.e.*, when reaching the next hop by following an energy weighted shortest path.

VII. CONCLUSIONS

In this paper we proposed Hypocomb family graphs and proved their planarity, connectivity and degree bound. Proofs omitted due to space limit can be found in [10]. This work opened a new line of research. Various follow-up works are possible. From theoretical point of view, it is an interesting topic to study the spanning ratio of Hypocomb family graphs, for example. In [10], we showed through a counter example that these graphs may not contain MST as subgraph. It is

therefore also interesting to study whether or not they are *low-weight* graphs. A structure is called low-weight if its total edge length is within a constant factor of the total edge length of the MST [14]. Another research topic is to develop graph construction algorithms more efficient than Algorithm 1.

In emerging wireless sensor and actor networks, Hypocomb and Reduced Hypocomb can be constructed among actors in a localized way, as Besh [11], by using directional message transmission to simulate ray drawing from each actor. Each node where blocking happens informs the sender actor about the blocking so that the latter knows about who it is blocking and whom it is blocked by. The goal is to obtain an actor overlay network bearing a planar topology so that existing data communication protocols can be run directly on it to realize, for example, actor-to-actor broadcasting, any-casting and multi-casting, which are central to actor-actor and sensor-actor coordination. However, this construction method does not produce exactly these graphs due to generally non-straight-line message transmission and thus inaccurate blocking relation, unless the underlying network has a grid topology. A future research direction is to study and improve the performance of this construction method and eventually develop new and better distributed/localized solutions. Comparative study of Local Hypocomb and other local planar graphs is also desired.

REFERENCES

- [1] P. Bose, L. Devroye, W. Evans, and D. Kirkpatrick. "On the spanning ratio of Gabriel graphs and beta-skeletons". In Proc. of LATIN, 2002.
- [2] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. "Routing with guaranteed delivery in ad hoc wireless networks". Wireless Networks, 7(6):609-616, 2001.
- [3] E. Elhasfi, N. Mitton, B. Pavkovic and D. Simplot-Ryl. "Energy-aware Georouting with Guaranteed Delivery in Wireless Sensor Networks with Obstacles." IJWIN, 16(3):142-153, 2009.
- [4] H. Frey and I. Stojmenovic. "On Delivery Guarantees of Face and Combined Greedy-Face Routing Algorithms in Ad Hoc and Sensor Networks". In Proc. ACM MobiCom, pp. 390-401, 2006.
- [5] J. Gao, L.J. Guibas, J. Hershberger, L. Zhang, and A. Zhu. "Geometric spanner for routing in mobile networks". Proc. MobiHoc, pp. 45-55, 2001.
- [6] F. Ingelrest, N. Mitton, and D. Simplot-Ryl. "A Turnover based Adaptive HELLO Protocol for Mobile Ad Hoc and Sensor Networks". Proc. MASCOTS, 2007.
- [7] J.W. Jaromeczyk and G.T. Toussaint. "Relative neighborhood graphs and their relatives". In Proc. of the IEEE, 80(9): 1502-1517, 1992.
- [8] J.M. Keil and C.A. Gutwin. "Classes of graphs which approximate the complete euclidean graph". Disc. and Comp. Geometry, 7:13-28, 1992.
- [9] N. Li, J.C. Hou, and L. Sha. "Design and Analysis of a MST-Based Topology Control Algorithm". In Proc. IEEE INFOCOM, 2003.
- [10] X. Li, N. Mitton, I. Simplot-Ryl, and D. Simplot-Ryl. "Graph Planarization in Wireless Ad hoc Networks." INRIA Research Report RR-7340, July, 2010. <http://hal.archives-ouvertes.fr/inria-00504683/fr/>
- [11] X. Li, N. Santoro, and I. Stojmenovic. "Localized Distance-Sensitive Service Discovery in Wireless Sensor and Actor Networks." IEEE Tran. on Computers, 58(9): 1275-1288, 2009.
- [12] X.-Y. Li. "Approximate MST for UDG Locally". In Proc. of COCOON, LNCS 2697, pp. 364-373, 2003.
- [13] X.-Y. Li, G. Calinescu, P. Wan, and Yu Wang. "Localized Delaunay Triangulation with Application in Ad Hoc Wireless Networks". TPDS, 14(10):1035-1047, 2003.
- [14] X.-Y. Li, Y. Wang, P.-J. Wang, W.-Z. Song and O. Frieder. "Localized low weight graph and its applications in wireless ad hoc networks". In Proc. of IEEE INFOCOM, 2004.
- [15] X.-Y. Li, Y. Wang, and W. Song. "Applications of k -Local MST for Topology Control and Broadcasting in Wireless Ad Hoc Networks". IEEE TPDS, 15(12): 1057-1069, 2004.
- [16] X.-Y. Li, I. Stojmenovic, and Y. Wang. "Partial Delaunay Triangulation and Degree Limited Localized Bluetooth Scatternet Formation". IEEE Tran. on Parallel and Distributed Systems, 15(4):350-361, 2004.
- [17] C. Monma and S. Suri. "Transitions in geometric minimum spanning trees". ACM Symposium on Computational Geometry, pp.239-249,1991.
- [18] Y. Wang and X.-Y. Li. "Localized Construction of Bounded Degree Planar Spanner for Wireless Networks". Mobile Networks and Applications, 11:161-175, 2006.
- [19] A.C.C. Yao. "On constructing minimum spanning trees in k -dimensional spaces and related problems". SIAM J. on Computing, 11:721-736, 1982.