# Local Computation and Global Communication in Ad Hoc Networks

By
Evangelos Kranakis

## Outline

- Ad Hoc Networking
  - Locality
  - Communication Paradigms
- Communication
- Simplifying the Infrastructure
- Channel Allocation
- Clustering

## Ad Hoc Networking

- There may not be any infrastructure:
  - remote areas
  - ad-hoc meetings
  - disaster areas
- Cost can be an argument against infrastructure:
  - battery resources
  - number of sensors
- Not every host can hear every other host:
  - Data needs to be forwarded in a "multi-hop" manner.

## Location Oblivious/Aware

- **Location Oblivious**
  - Hosts have no knowledge of network layout.
  - Is this assumption "too minimalist"?
  - How can effective communication principles be established?
- **Location Aware**
  - Overhearing, Localization, GPS, etc., are often available.
  - Simple communication exchanges can be used to establish an underlying topological structure.
  - Can this be used to our advantage?

**Question:** How location oblivious/aware should one be?

## Locality

*Locality in Distributed Graph Algorithms (Linial, 1992)*

- A distributed algorithm is called **local** if each node of the network makes decisions based only on information obtained from nodes located no more than a **constant** (independent of the size of the network) **number of hops** from it.
- **Not the same as distributed:** No host is ever aware of the existence of the parts of the network further away than this constant number of hops.

There are several reasons why such local algorithms are practical for wireless, ad hoc networks.

## Importance of Locality

1. Changes in the network outside a constant-size neighborhood do not influence the computation.
2. Adapting to a change in the network requires solely a local recalculation of the solution.
3. It is possible to calculate only a part of the required subnetwork that is really needed without necessarily having to calculate a complete solution (this can be important in cases of disaster recovery).
4. Messages do not propagate indefinitely throughout the network and the algorithm terminates in a constant number of steps.
5. A solution is consistent regardless of the order in which the nodes or edges are considered in the calculations.

## Location Awareness and Locality

An important question:

- Given the limitations of locality, and
- Assuming location awareness
- Can we improve communication efficiency in ad hoc networks?

## Communication Paradigms

- Two important paradigms are being used in “location aware” ad hoc networks.
  - **Local → Global**
  - **Global → Local**
- **Local → Global**
  An algorithm is *suitable* for local computation provided that it attains *good* global connectivity and *spanning* characteristics.
- **Global → Local**
  Use global algorithms restricted to a geographically local vicinity.

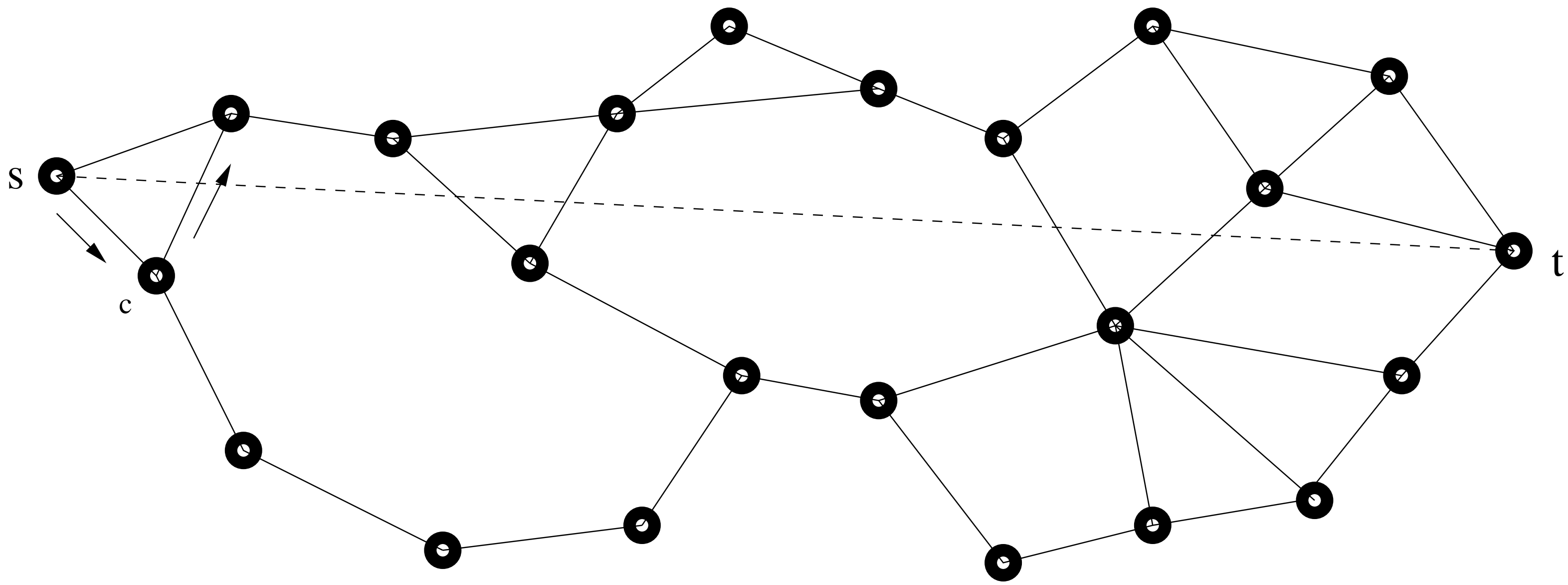
# Communication

## Face-Routing in Planar Networks

**Routing from $u$ to $v$.**

1. Starting at $c := u$ determine face $F := F_0$ incident to $c$ intersected by the line segment $\vec{st}$.
2. Select any of the two edges of $F_0$ incident to $c$ and start traversing the edges of $F_0$ until we find the second edge, say $xy$, of $F_0$ intersected by $\vec{uv}$.
3. Update face $F$ to the new face of the graph containing edge $uv$, and vertex $v$ to either of the vertices $x$ or $y$.
4. Iterate until $v$ is found.

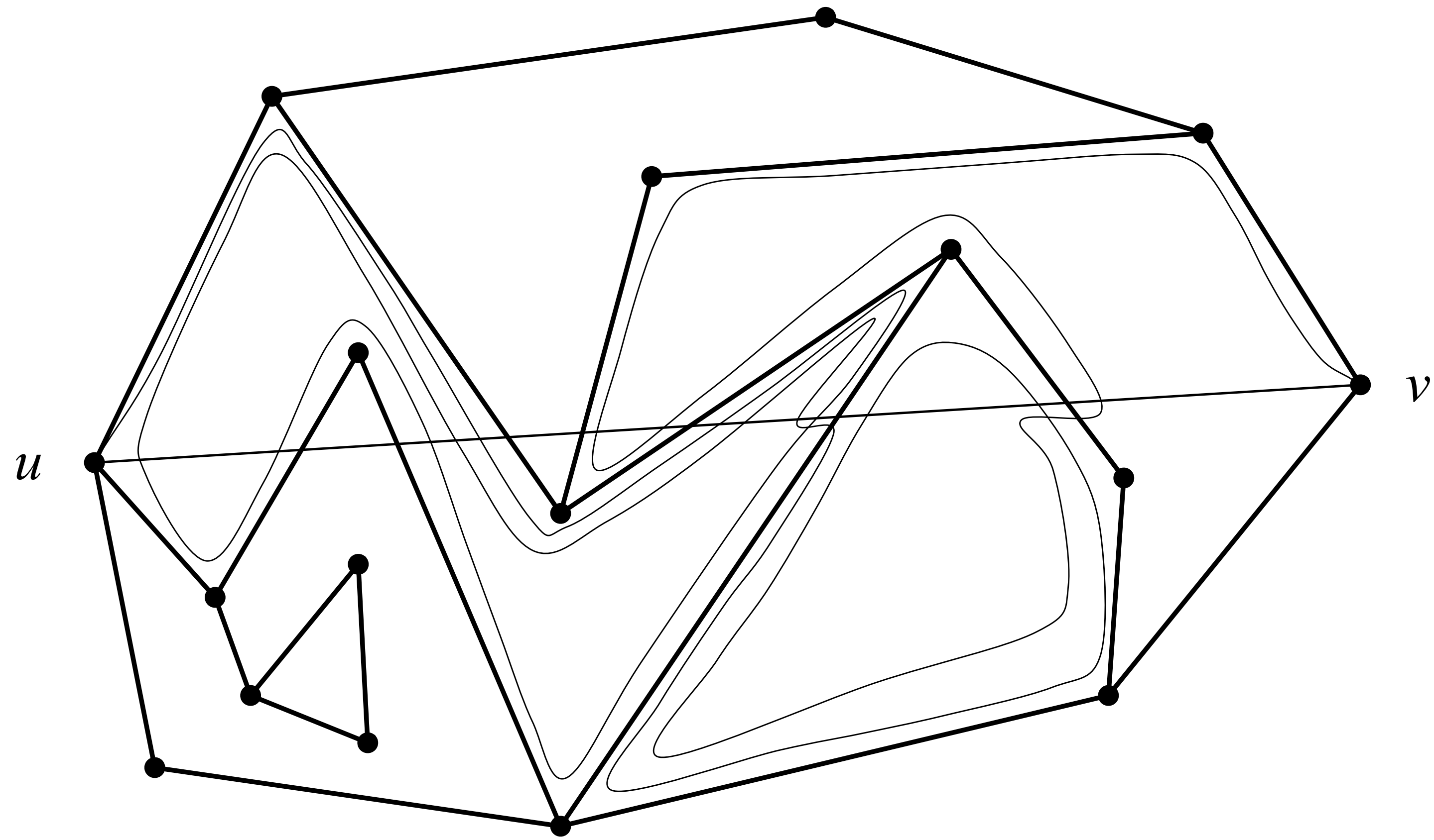
# Routing from $s$ to $t$ (Example 1)




Initially $c := s$.

Update $c$ and repeat.

## Face Routing (Example 2)



## Properties

**Source must have knowledge (geographic coordinates) of destination.**

- It is a local algorithm.
- It is guaranteed to succeed.

**Unfortunately, it works only for planar graphs!**

# Simplifying the Infrastructure

## Why Simplify the Infrastructure?

- Remember,...,we want to establish efficient communication in a "locality setting" meaning we do not have global knowledge of the ad hoc network!
- How do you establish communication (e.g., routing) with guaranteed delivery? Can do it with flooding...but doesn't seem to be a good idea!
- **Accepted method:** Find a planar spanner of the original ad hoc network.
- **Question:** But how do you find a good spanner satisfying the "locality" requirement?

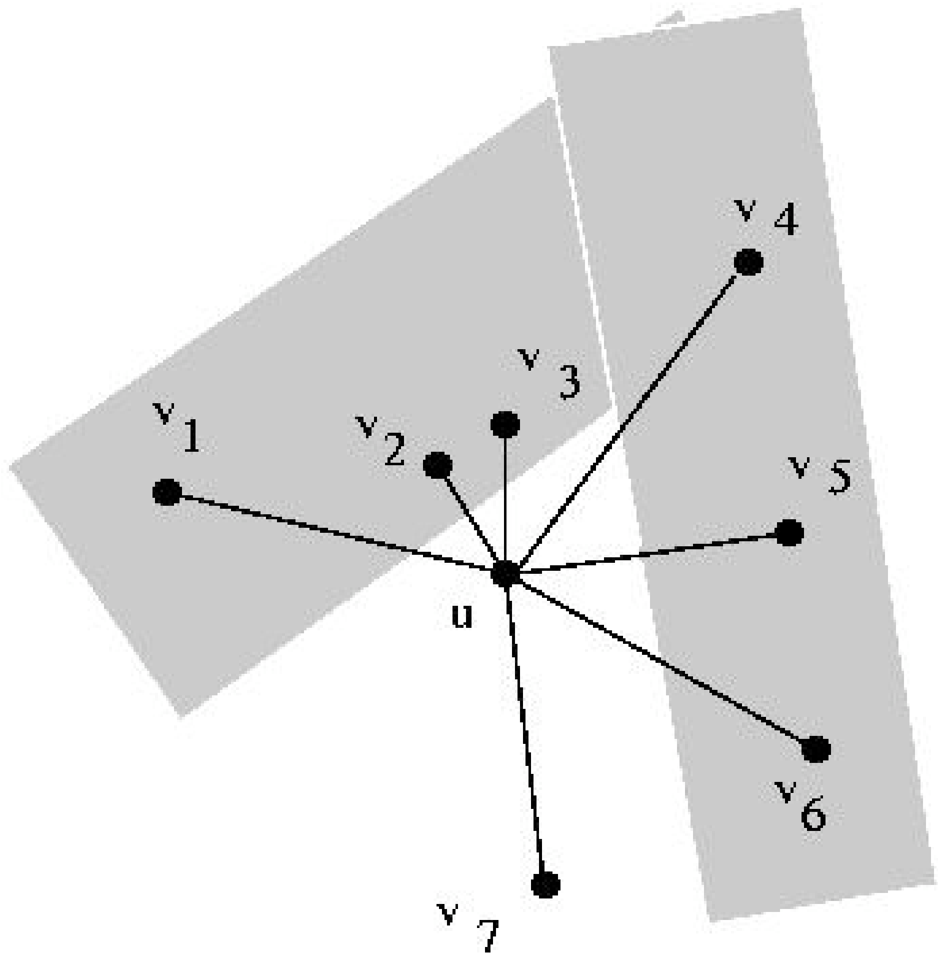
## Half-Space Proximal: Idea

This test resembles the Yao test in that each point $u$ patitions the plane around it (in this case in two parts).

The idea is the following.

1. $u$ selects the point closest to it, say $v$.
2. $u$ draws the line $L$ perpendicular to $uv$ at its midpoint.
3. Removes from consideration all vertices in the half-plane determined by $L$ that contains $v$.
4. Iterate.

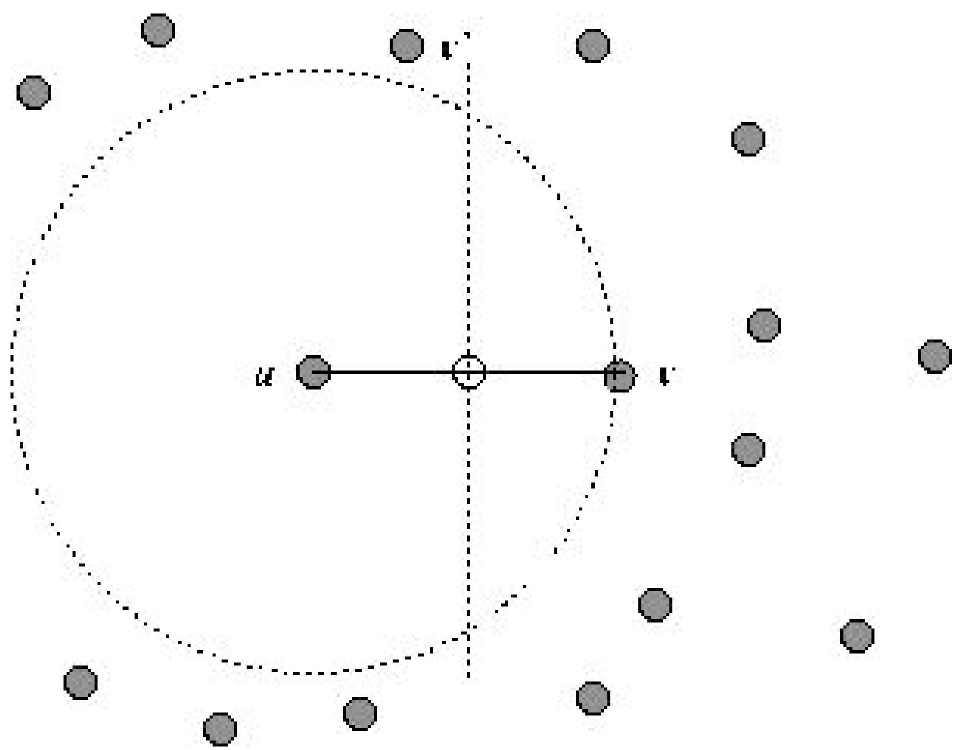
## Example




$u$ selects edges $uv_2, uv_5, uv_7$.

## Bounded Degree $\leq 6$

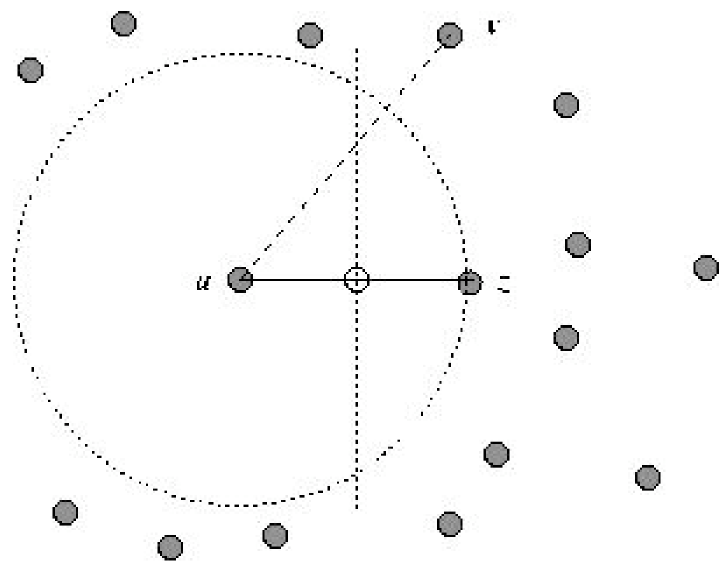
Let $uv$ be the shortest edge from $u$ and let $v'$ be next shortest edge.



The angle $\angle vuv'$ must be $\geq \pi/3$,

## Strongly Connected

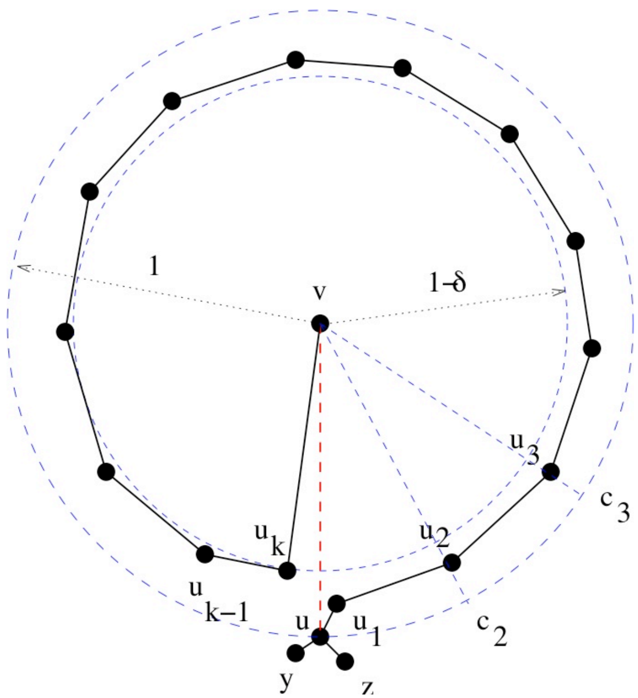
Let $uv$ be the *shortest counterexample* of an edge of $G$ with no directed path from $u$ to $v$. By construction, there must exist an edge $uz$ selected and $v$ is in the forbidden region.



Therefore $\angle vuz \geq \pi/3$. However, $|zv| < |uv|$. Since $G$ is a UDG, $zv$ must be an edge of $G$. Since $uv$ is the shortest counterexample there must exist a directed path from $z$ to $v$. **Contradiction!**

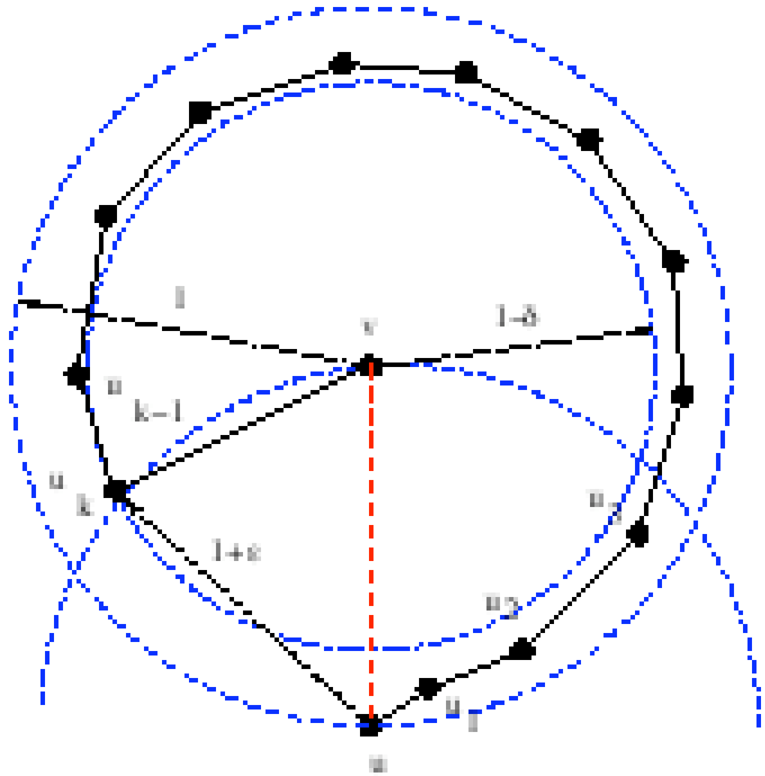
## Stretch Factor $\leq 2\pi + 1$

Suppose that an edge $uv$ of length $r \leq 1$ is excluded by $u$.




This means that there is an edge $uu_1$ in $G$ which is selected by $u$ such that $|uu_1| \leq |uv|$ and $\angle u_1uv < \pi/2$. Thus the edge $uu_1$ makes the vertex $v$ to be in the forbidden area. If the edge $u_1v$ is selected then the stretch factor is less than 3, else we can argue inductively that there exists a sequence of vertices $u_0 = u, u_1, u_2, u_3, \ldots, u_{k+1} = v$ such that ... Look in the paper for the proof!

## Spanner with Dilation $5\pi/3+1$




Look in the paper for the proof.

## Half-Space Proximal

1. If $G$ is a connected UDG then the digraph $\overrightarrow{HSP}(G)$ has out-degree at most 6 (can be made to have degree at most 5) and is strongly connected.

2. Let $G$ be a geometric UDG and $\overrightarrow{HSP}(G)$ be the digraph constructed from $G$ by the above algorithm. Then the stretch factor of $\overrightarrow{HSP}(G)$ is at most $2\pi + 1$.

3. If $G$ is a connected unit disk graph then a geometric minimum spanning tree of $G$ is a subgraph of $HSP(G)$.

4. The Half-Space Proximal is not necessarily planar.

## From Algorithm to $k$-Local-Algorithm

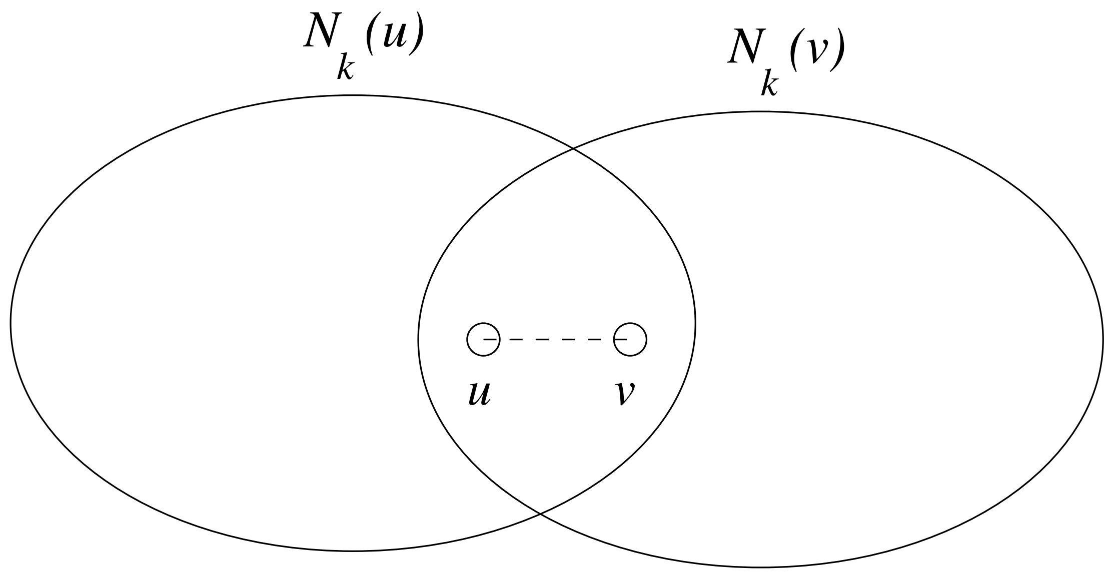
Take any algorithm $\mathcal{A}$ such that on input a UDG $G$, the algorithm outputs a spanner $\mathcal{A}(G)$ of $G$.

**Algorithm $k$-Local.**
**Input:** $G$.
**Output:** $G'$.

1. For any node $u$,
   (a) $u$ collects its distance $k$ neighborhood $N_k(u)$.
   (b) $u$ constructs $\mathcal{A}(N_k(u))$.
2. A link $\{u, v\}$ is accepted in $G'$ if it is in both $\mathcal{A}(N_k(u))$ and $\mathcal{A}(N_k(v))$.

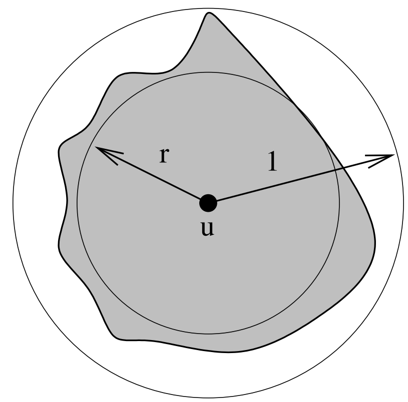
Changes to (2) are also possible.

# From Algorithm to $k$-Local-Algorithm



## UDGs with Irregularity $r$

Two nodes at distance at most $r$ can communicate directly, but no nodes at distance more than 1 can communicate directly.




Nodes $u, v$ may or may not communicate directly if $r < d(u, v) \leq 1$.

For a given a set $P$ of points in the plane and for each $r$, $UDG(P; r)$ is the resulting class of graphs.

Each $G \in UDG(P; r)$ contains (resp., is contained in) the $UDG$ on $P$ with radius $r$ (resp., 1).

## $k$-Local Spanning Trees

**Algorithm: LocalMST$_k$**

**Input:** Connected geometric graph $G$ with the linear order $\prec$;

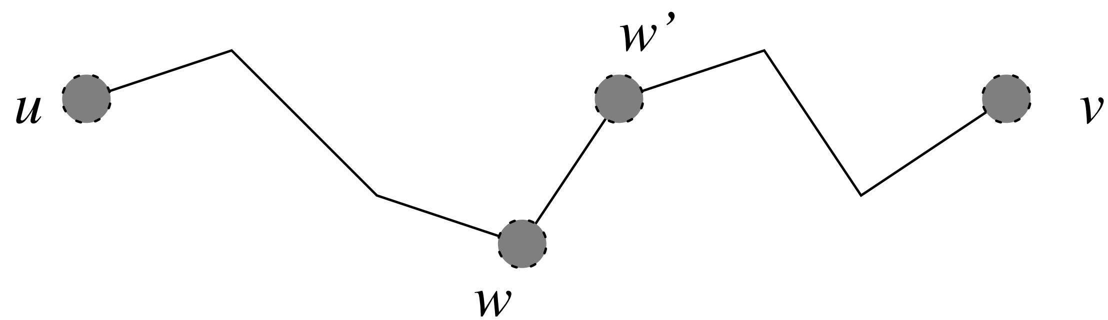
**Output:** Graph $G_k^{\prec}$

Run the following algorithm at each node $v$ of $G$:

1. Learn your distance $k$ neighborhood $N_k[v]$.
2. Construct locally the unique MST $T_k(v)$ of $N_k[v]$.
3. Broadcast in $N_1[v]$ the edges of $N_1[v]$ which have been retained in $T_k(v)$ (i.e. $N_1[v] \cap T_k(v)$).
4. The output graph $G_k^{\prec}$ is defined as follows: an edge is selected into $G_k^{\prec}$ if and only if it was retained by both of its incident nodes.

## Connectivity

Assume on the contrary $\{u, v\}$ is retained in MST $T$, but rejected in $G_k^{\prec}$ (wlog assume rejected in $T_k(v)$): there exists a path, say $p$, in $T_k(v)$ joining $u$ and $v$ and using only edges smaller than $\{u, v\}$.

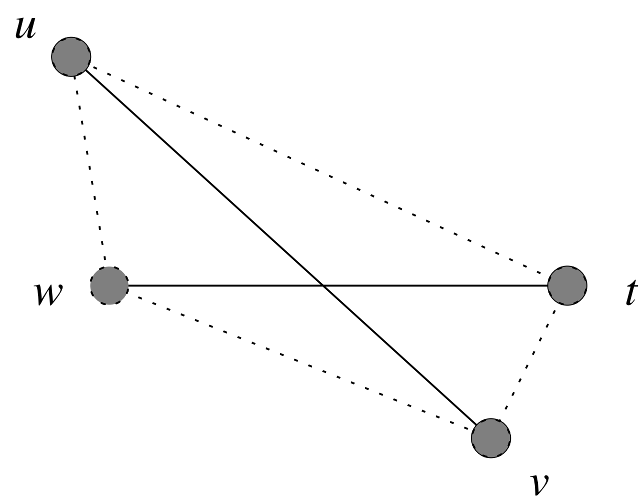



Let $\{w, w'\}$ be an edge in $p$ such that $\{w, w'\} \notin T$: hence, there is a path in $T$ joining $w$ and $w'$ and using only edges smaller than the edge $\{w, w'\}$.

This contradicts the fact that the edge $\{u, v\}$ was retained in $T$.

## Planarity, if Min distance $\geq \sqrt{1-r^2}$ (1/2)

Assume on the contrary $G_k^{\prec}$ not planar and $\{u,v\},\{w,t\}$ be two crossing edges in $G_k^{\prec}$ (wlog, largest angle in quadrilateral, say $\angle uwv$, is $\geq \pi/2$).

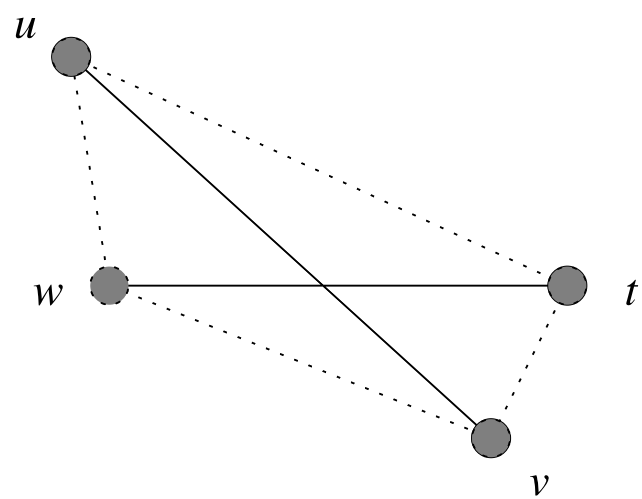



We have $|u,w|^2 + |w,v|^2 \leq |u,v|^2 \leq 1$. Since $|w,v| \geq \sqrt{1-r^2}$, $|u,w|^2 \leq 1 - |w,v|^2 \leq r^2$. Hence, $\{u,w\} \in G$. Similarly, $\{w,v\} \in G$.

**Claim:** $\{u,v\}$ will not be selected into $G_k^{\prec}$ by $u$.

## Planarity, if Min distance $\geq \sqrt{1-r^2}$ (2/2)

Node $u$ computes $T_k(u)$ using Kruskal's algorithm. Either $\{u, w\}$ is retained in $T_k(u)$, or there already exists a path in $T_k(u)$ consisting of smaller edges connecting $u$ and $w$. Same is true for $\{w, v\}$.




So when $\{u, v\}$ is being considered by $u$ for inclusion into $T_k(u)$, there already exists a path in $T_k(u)$ connecting $u$ and $v$ and hence $\{u, v\}$ will be rejected by $u$, which contradicts the fact that edge $\{u, v\}$ is in $G_k^{\prec}$.

## $k$-Local Spanning Trees

- The resulting spanner has maximum degree at most $3 + \frac{6}{\pi r} + \frac{r+1}{r^2}$, when $0 < r < 1$ (and at most five, when $r = 1$).
- The spanner is *planar* provided that the distance between any two nodes is at least $\sqrt{1 - r^2}$.
- For $k \geq 2$ the sum of the euclidean lengths of the edges of the spanner is at most $\frac{kr+1}{kr-1}$ times the sum of the euclidean lengths of the edges of a minimum weight euclidean spanning tree if the spanner is planar.