# Topology control:

Sometimes network can be too dense, i.e, many nodes in close vicinity. Therefore, topology control methods deal with such networks by reducing / controlling transmission power, deciding which links to use & turning off few nodes.

**Motivation:** (Dense ntwk provide better connectivity & handles node failures)
In dense n/ork, too many nodes leads to too many collisions/ too complex operation for MAC protocol, too many paths to chose from with routing protocols. More Tx^n Power (Bandwidth) required for nodes to communicate distant nodes.

The idea is to make topology less complex by mentioning which node is able/allowed to communicate with other nodes.

**Topology control** (Topology control algorithm takes the graph $G=(V,E)$ representing a n/ork and transfers it to graph $T=(V_T, E_T)$ such that $V_T \subseteq V$ & $E_T \subseteq E$

- Control node activity → deliberately turn on/off nodes
- Control link activity → deliberately use/not use certain links

**Topology control**

- Flat network - all nodes have essentially same role
  - Power control (using power modulation)
- Hierarchical network - assign different roles to nodes: exploit that to control node/link activity.
  - Backbones
  - Clustering

**Flat networks:**

These networks mainly control transmission power. This is achieved by
- → Not always using maximum power
- → selective for some links or for a node
- → makes topology thinner
- → less interference

The alternation to this is to selectively discard some links.

**Hierarchical networks - backbone.**

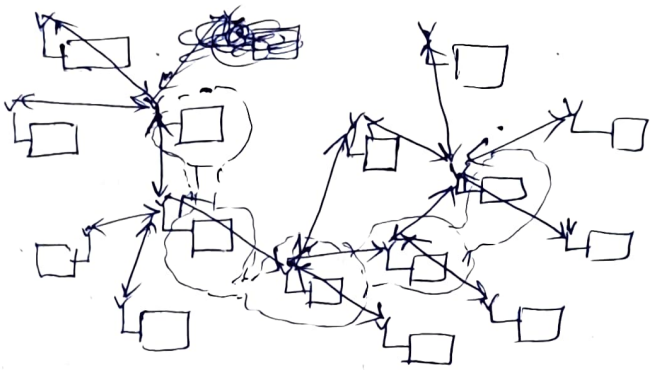
Here a backbone network is constructed and only the links in backbone and from backbone to controlled neighbors are used.

In this network some nodes control their neighbors and form a dominating set.

→ Each node have a controlling neighbor

→ Controlling nodes have to be connected (backbone)

→ only links wether backbone & from backbone to controlled neighbors are used.



## Hierarchical network (Clustering)

→ Here nodes are partitioned into clusters. (clusters are formed if certain conditions are hold)

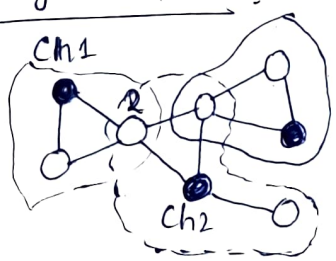
→ A specific node is with in one group only except for bridging node bet^n two or more groups.

→ Groups also have cluster-heads

→ Clusterheads are a dominating set but separated from each other, ie they form independent set.

→ In a cluster each node is one hop away from clusterhead. This is the basis of forming clusterhead.

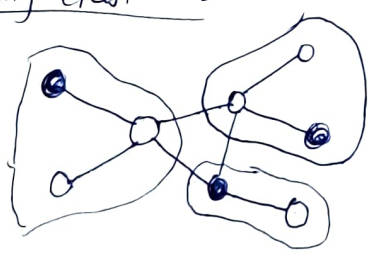
## Overlaping clusters :




In this scenario node 2 is adjacent to clusterheads 1 & 2. Therefore, this node is assigned to both clusters resulting in overlaping clusters. TO avoid this some decision is made to unambiguously assign node to clusterheads.

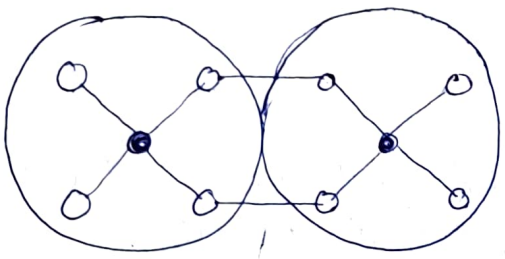
Soln:

## Non overlapping clusters



How do clusters communicate:

Irrespective of cluster overlapping, a node which is adjacent to two clusterheads provide the comm^n. between two clusters and act as a gateway. Hence, intracluster comm^n can be routed via clusterheads. If two clusterheads are separated by two nodes, in this case the two nodes from each cluster can act as distributed gateway to facilitate comm^n. bet^n clusters.



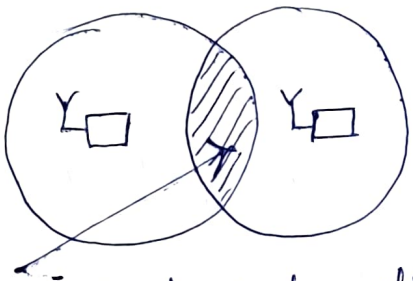
→ Maximal diameter of a cluster: The max^m. cluster diameter of two cells each node in cluster is at most two hops away from any other node. Multihop clusters use larger diameter.

## Some examples constructions for power control

The basic idea for most methods is take a graph $G = (V, E)$ produce a graph $G^{\circ} = (V, E^{\circ})$ that maintains connectivity with fewer edges. It can be achieved through
→ Assume the knowledge about node positions.
→ construction should be local.

Some examples are : Relative Neighborhood Graph (RNG), Gabriel graph, Delaunay triangulation

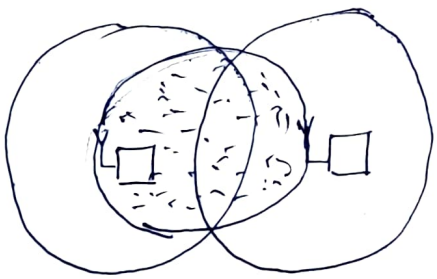
### RNG :




This region has to be empty for the two nodes to be connected

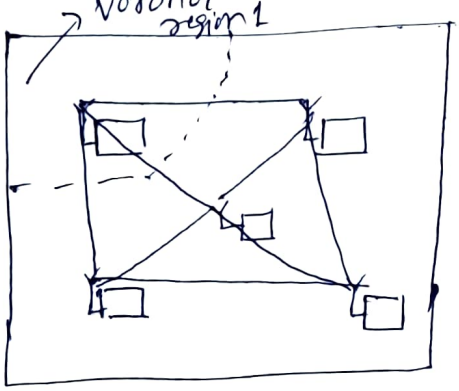
→ Edge betn nodes u & v if and only if there is no other node w that is closer to either u or v.
→ RNG maintains original graph connectivity.
→ Easy to compute locally.

### Gabriel graph



→ Similar to RNG
→ the smallest circle with nodes u & v on its circumference must only contain node u & v for u & v to be connected.

### Delaunay triangulation :




→ Voronoi diagram is a collection of Voronoi polygons, which are formed around each point that includes all points closer to it than any other point in the set.
→ Delaunay triangulation connects any two nodes for which Voronoi regions touch.

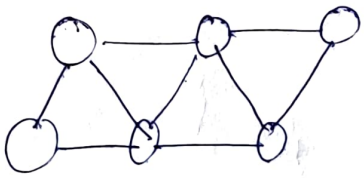
# Hierarchical networks – backbone

## Backbone by growing tree

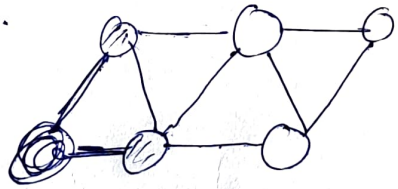
→ Initialize all nodes color to white, Pick an arbitrary node & color it gray.

While (there are white nodes) {
  pick a gray node v that has white neighbors color the gray node v black
  for each white neighbor u of v {
    color u gray
    add (v, u) to tree T.
  }
}

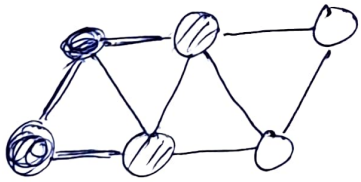
1.



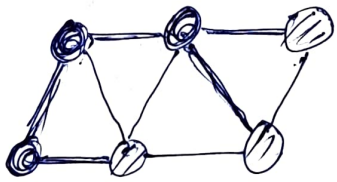
2.



3.



4.



only solid links are considered links.