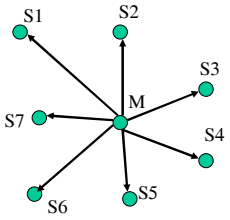
Bluetooth scatternet formation

Tutorial 3

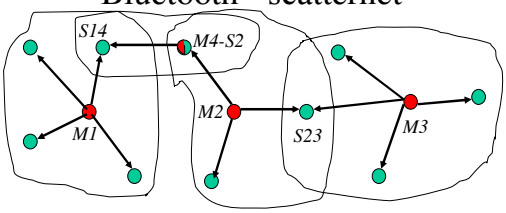
Ivan Stojmenovic
Ivan@site.uottawa.ca
www.site.uottawa.ca/~ivan

Bluetooth - piconet

- Short-range
- Master-slave
- Frequency hopping
- PICONET = master + K slaves, K≤7
- Additional slaves must be parked



Bluetooth - scatternet



Connect piconets into scatternet

Avoid master-slave bridges

Bridges participate in piconets on time division basis

Minimize number of slave roles and # of masters

Scatternet in single-hop networks

- Salonidis, Bhagwat, Tassiulas, LaMaire 2001: centralized, max 36 nodes
- Law, Mehta, Siu 2001: growing one tree
- Tan et all MIT 2001: growing trees
- Sun, Chang, Lai 2002: search trees; Bluetooth IDs as keys; overhead for scatternet merge/maintenance
- Cgun-Choong, Kee-Chaing 2002: Bluerings
- Baatz, Bieschke, Frank, Martini, Scholz 2002: 1-factors, piconets not degree limited
- Miorandi, Zanella 2002: centralized, traffic known a priori
- Barriere, Fraignaud, Narajanan, Opatrny 2003: Projective geometry, connected degree limited

Large overhead for construction and maintenance in all cases

Scatternets for multi-hop networks

- Growing trees (overhead)
  Ramachandran, Kapoor, Sarkar, Aggarwal 2000
  Zaruba, Basagni, Chlamtac 2001
- Clustering based, no degree limit or not connected:
  Basagni, Chlamtac, Petrioli 2001
  Balaji, Kapoor, Nanavati, Ramachandran 2001
  Wang, Thomas, Haas 2002 Guerin, Kim, Sarkar 2002
- Guerin, Sarkar, Vergetis 2002: DFS, BFS, MST 2-D, 3-D
- Ajmone-Marsan, Chiasserini, Nucci, Carello, Giovanni 2002: centralized, traffic known in advance
- Clustering based: connected, piconets degree limited, but # of slave roles not limited: Petrioli and Basagni 2002
- + Position information, connected, fully degree limited
  Li and Stojmenovic 2001 Stojmenovic 2002

Scatternet by growing tree

- Ramachandran, Kapoor, Sarkar, Aggarwal 2000: grow tree from root, master not always directly connected to its slave
- Zaruba, Basagni, Chlamtac 2001: grow tree from root, at most 5 slaves per master; if >5, select two connected slaves, link them, and disconnect one;
  Multiple blueroots extension
- *Communication overhead and Scatternet maintenance?*
- *Tree has root as communication bottleneck*

Clustering based scatternet formation

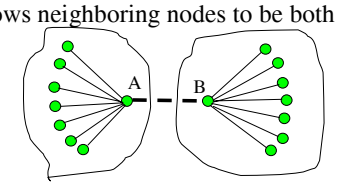
- Basagni, Chlamtac, Petrioli 2001
- Detect neighboring nodes by paging and scanning
- Apply clustering process
- Clusterheads = masters
- Nodes in a cluster = slaves
- Connect clusters = bridge piconets
- *degree (number of slaves) not limited to 7*
- parking and unparking process ?
- Maintenance is not localized – chain effect ?

Neighbor discovery in Bluetooth

- Each node decides with probability 0.5 between inquiry and inquiry-scan modes
- Senders and receivers change frequencies in mutually random pattern (? 32 frequencies)
- If sender and receiver are on the same frequency at some time, they discover each other, and establish master-slave relation
- In multi-hop networks, overall connectivity established quickly, but full awareness of all neighbors is slow

Degree limited clustering based

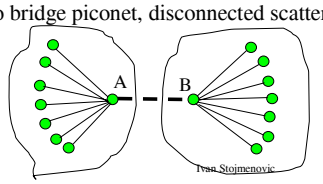
- Balaji, Kapoor, Nanavati, Ramachandran 2001
- Master = Highest degree node (degree for undecided neighbors only), up to seven neighbors with smallest degree are slaves
- Allows neighboring nodes to be both masters



No bridge piconet, disconnected scatternet ?

Scatternet by random key clustering

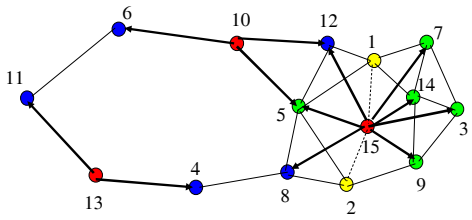
- Wang, Thomas, Haas 2002
- Guerin, Kim, Sarkar 2002
- Node decides to be master at random
- and then 'slaves' up to seven neighboring nodes
- Connect scatternet by bridge piconets
- No bridge piconet, disconnected scatternet ?



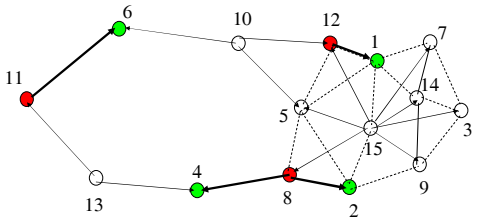
Connected out-degree limited scatternet

- Petrioli and Basagni 2002: **BlueMesh**
- Two rounds discovery phase for 2-hop information
- If a master has more than 7 neighbors, it chooses up to 7 among them so that it can reach others via them
- Scatternet formation in clustering based iterations:
  initially all nodes undecided
  init nodes (biggest weight among undecided neighbors) create piconets, delete unused edges, and designate some slaves for links to other piconets
  stop when all nodes decided, withdraw masters and undesignated slaves from the next iteration
- Connected but # of slave roles not limited

BlueMesh: iteration 1



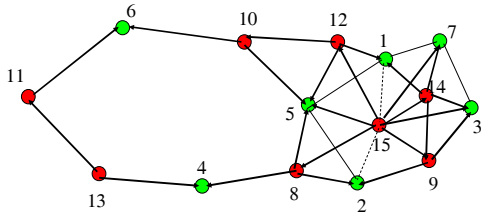
BlueMesh: Iteration 2



MIS based BSF (Stojmenovic)

- MIS = maximal independent set: each node is in MIS or is a neighbor of node from MIS, no two nodes from MIS are neighbors
- Finding MIS: repeatedly pick up a neighbor A and eliminate all common neighbors with A
- Iteration 1: each node creates piconet by selecting MIS of its neighbors as slaves, plus adding more up to 7
- Iteration 2: Following clustering methodology, each node decides whether or not its piconet is needed for connectivity
- Same problem: # of slave roles is not limited

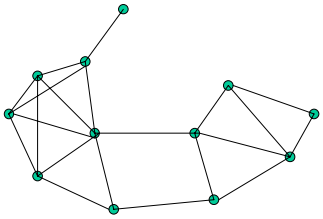
Example



Degree limited connected scatternet formation

- Li, Stojmenovic, Wang 2001 – clustering based
- Stojmenovic 2002 – dominating set based
- *Phase I* = create unit graph and (optional) construct a planar connected structure in localized manner
- *Phase II* = eliminate some edges in the structure to **limit the degree** of each node to 7
- *Phase III* = **decide master-slave roles** between two nodes of each edge in the structure
- Phases II and III **iterative** (node by node) or concurrently by all nodes **simultaneous** approach)
- Phase III **during** or **after** Phase II

Scatternet formation – optional phase I



Unit graphs
radius

Create unit graph

Assumption: Each node is aware of its position and learns position of all neighbors within transmission radius

Construct **planar connected** structure in localized manner:
MST ⊆ **LMST** ⊆ **RNG** ⊆ **GG** ⊆ **PDT** ⊆ DT

Sparse graphs: LMST and RNG

LMST – each node constructs Local MST (from local edges); keep edge if it belongs to MSTs of both endpoints

RNG – Relative Neighborhood Graph (Toussaint 1980): Keep edges which are not longest in any triangle

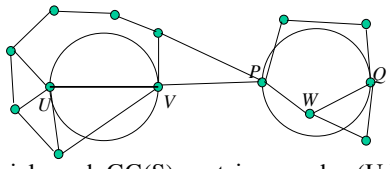
Unique edge weights:
$w(AB)=(|AB|, min(ID(A), ID(B)), max(ID(A), ID(B)))$.

Each node in LMST and RNG with unique edge weight has degree < 6

RNG has average degree < 2.4 = sparse

LMST has average degree 2.04, < 3% more edges than MST

Gabriel graph



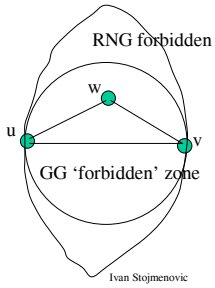
Gabriel graph GG(S) contains an edge (U,V)
iff the disk with diameter (U,V)
contains no other point from S

Computing GG from unit graph requires no message exchange

Some nodes may have high degrees

RNG ⊆GG

- $uv \in RNG \rightarrow$ no w, $uw < uv$, $vw < uw \rightarrow$ disk with diameter uv has no node $\rightarrow uv \in GG$

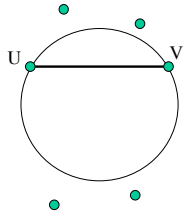


Partial Delaunay Triangulation

Li, Stojmenovic 2001

Delaunay Traingulation = dual Voronoi diagram

UV in DT iff there exist a circle with chord UV without other nodes inside it



Test disk with diameter UV:

If empty then UV in PDT

If nodes inside disk on both sides then not in PDT

Find smallest angle on both sides of UV

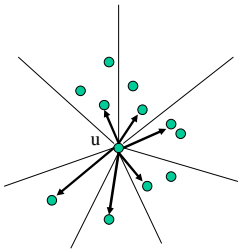
If together $\geq \pi$ then not in PDT

If together $< \pi$ then in PDT iff both are neighbors, using 1-hop or 2-hop info

PDT= portion of DT which can be decided locally

Yao graph – Phase II

k=7

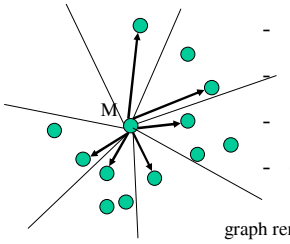


Divide into k equal cones around u

Find closest point in each cone, if any

Limiting degrees – cluster based

Phase 2: Applied on **active** nodes = nodes with highest keys among undecided neighbors



- apply Yao construct
- assign role to itself
- marks itself decided
- contacts neighbors about deleted edge or its role

graph remains undirectional

Deciding master-slave roles

-Key= Bluetooth ID (one-hop neighbor discovery)

-Key= (degree, ID) (two-hop neighbor information)

-Active node decides on roles on each undeleted edge:

-Higher original key, or

-Clustering based: if undecided or any master role assigned → master role, otherwise slave role to itself

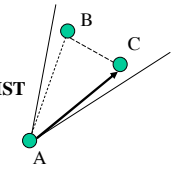
-Variant: if any slave role → slave role

Yao construct preserves connectivity

•Sort all edges of unit, GG or RNG or PDT by
w(AB)=(|AB|, min(ID(A), ID(B)), max(ID(A), ID(B))).

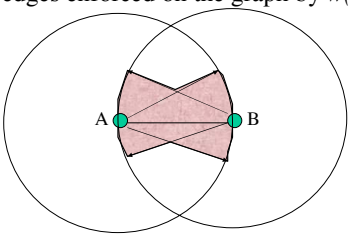
•Construct MST by considering edges in increasing order, include if no cycle created (Kruskal's scheme)

If AB not in Yao then ∃ C, AC in Yao,
AB < AC,
BC < AB (angle < π/6)
⇒ BC, AC already considered for MST and connected,
⇒ AB not needed in MST
⇒ MST is included in Yao



Yao bidirectional edges is connected

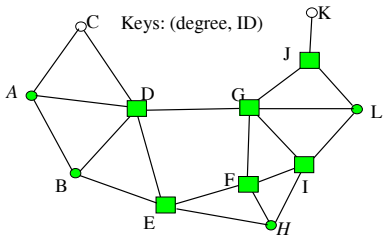
- Contains RNG as subgraph (Wang, Li 2002)
- Unique edges enforced on the graph by *w(AB)*



Dominating set based master-slave roles

- Phase 2 applied on each node simultaneously
- Phase 3: assigning roles to endpoints of each edge
- Localized maintenance
- Solution 3:
  find a dominating set,
  key=(dominating, degree, id)
  higher key node on each edge is master
  Two non-dominating nodes may not be connected
  Nodes in dominating set only can be masters

Dominating sets



Wu, Li '99

Higher degree priority: [SSZ]

Localized maintenance;

No communication overhead

C,K not intermediate = any two neighbors connected

A,B,H,L not inter-gateway = covered by a neighbor

Any path via H can be replaced by a path via F (EHI → EFI)

Gateway nodes = not covered by two connected neighbors

Avoiding position information ?

- Single-hop networks: **virtual positions** (Wang, Li, Stojmenovic 2004)
- MST can be used instead of LMST, RNG, GG
- Delaunay triangulation can also be used (with Yao structure) for denser planar graph

On-demand scatternet formation

- Liu, Lee, Saadawi 2003
- Create scetternet only when route is needed
- Use flooding, which creates piconets during destination search, or
- Flooding could use single channel MAC while routing later uses scatternet/Bluetooth
- Destroy scatternet when routing completed

Three-dimensional scatternets ?

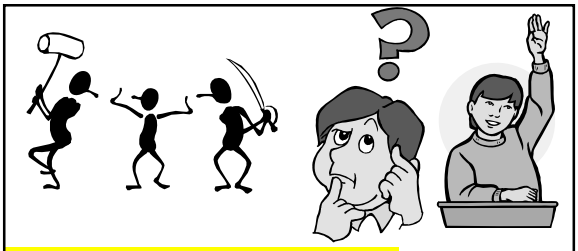
- Guerin, Sarkar, Vergetis 2002: DFS, BFS, MST
- They construct a tree where all nodes at one level are either masters or slaves (bipartite graph)
- MST nodes in 3-D can have degrees up to 13
- MST not localized, but RNG contains MST

Optimize edge lengths in scatternets

- Use sparse localized network (RNG or LMST)
- Degree of each node already limited to 6, average is 2 (LMST)
- Selected edges are *short* edges
- Add long edges up to degree 7, choose them in all directions
- Routing has reduced hop count due to adding four long edges on average
- For power efficiency add edges of preferred length

Future work

- Experiments: not many
- Bluetooth scatternet formation without position information and with slave degree limitation ??
- Routing in scatternets
- Power efficient scatternets
- Denser planar graphs ?
- Neighbor discovery and non-unit graphs
- Scheduling, capacity, QoS …
- Fuzzy unit graph for scatternet formation
- Three-dimensional scatternets ?



- *Ivan Stojmenovic*
- ivan@site.uottawa.ca
- www.site.uottawa.ca/~ivan