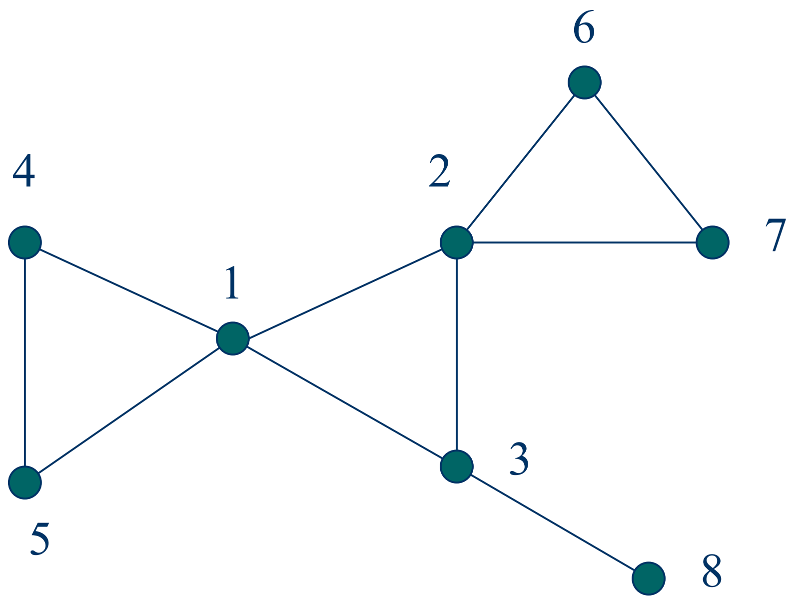
Ad Hoc Multipoint Communication

Stefano Basagni
Dept. of Elec. And Comp. Eng.
Northeastern University

A Simple Ad Hoc Network



Technical Challenges for the Ad Hoc Architecture

- Given **node mobility**: Every node in the network can move unpredictably and independently, at variable speed
- Given **a very large number of nodes**: For supporting pervasive computing, sensing of large geographic areas, etc.
- Given the nodes' **limited resources**

➢ Network protocols need to be **robust**, **reliable** and **scalable** (which makes the network such)

Multipoint Communication

The most general form of communication. It includes:

- One-to-one (routing)
- One-to-many (multicast)
- One-to-all (broadcast)
- Many-to-many (gossiping)

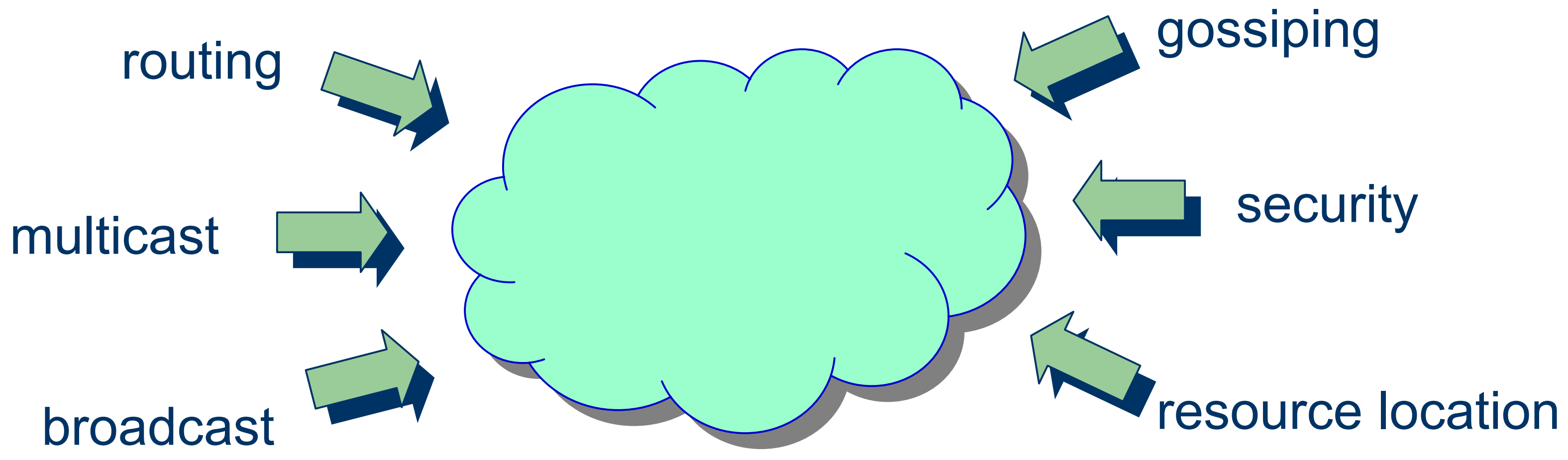
Multipoint Communication-Cont.

Previous approaches:

- Focus on a single communication problems
- Several diverse techniques (not efficiently "reusable")
- Have to maintain communication information at each node (state information)
- Devote lots of resources to control traffic

A Unifying Approach

- An architectural concept that implements network services and communication protocols without maintaining communication information at the nodes.



Inside the Cloud

- A:
  - Fast and simple
  - Resource efficient
  - Mobility adaptive
  - Node-status dependent

  Node selection mechanism to efficiently select and maintain ONLY a small subset of nodes for implementing network services and protocols

How to Select the Best Nodes

- Independence of the clusterheads
- Dominance of the clusterheads
- Possibility to express “preferences”
- Distributed operations
- Fast and simple implementation

Previous approaches

- Heuristics based on Independent Sets
  - Minimum ID approach (Gerla & al.)
  - Maximum degree (Ephremides & al.)
- Heuristics based on Dominating Sets
  - The concept of "spine"
  - Minimum connected dominating set

Previous approaches: Drawbacks

- No preferences
- Clustering “set up” differs from clustering maintenance
- One *and* two hops neighbors have to known at each node
- Problems with nodes mobility
- No analytical results

Our Approach: Maximal Weighted Independent Set based Clustering

- Clustering selection based on generic **weights** (real numbers > 0)
  - Mobility/node related parameters
  - Generalizes previous “Independent Set” solutions

Two algorithms

- Distributed Clustering Algorithm (DCA)
  - Quasi-mobile networks, periodical reclustering. Allow complexity analysis, fast and simple
- Distributed and Mobility-Adaptive Clustering (DMAC) Algorithm
  - Same rules/procedures for clustering set up and maintenance, adaptive to nodes mobility and node/link failures

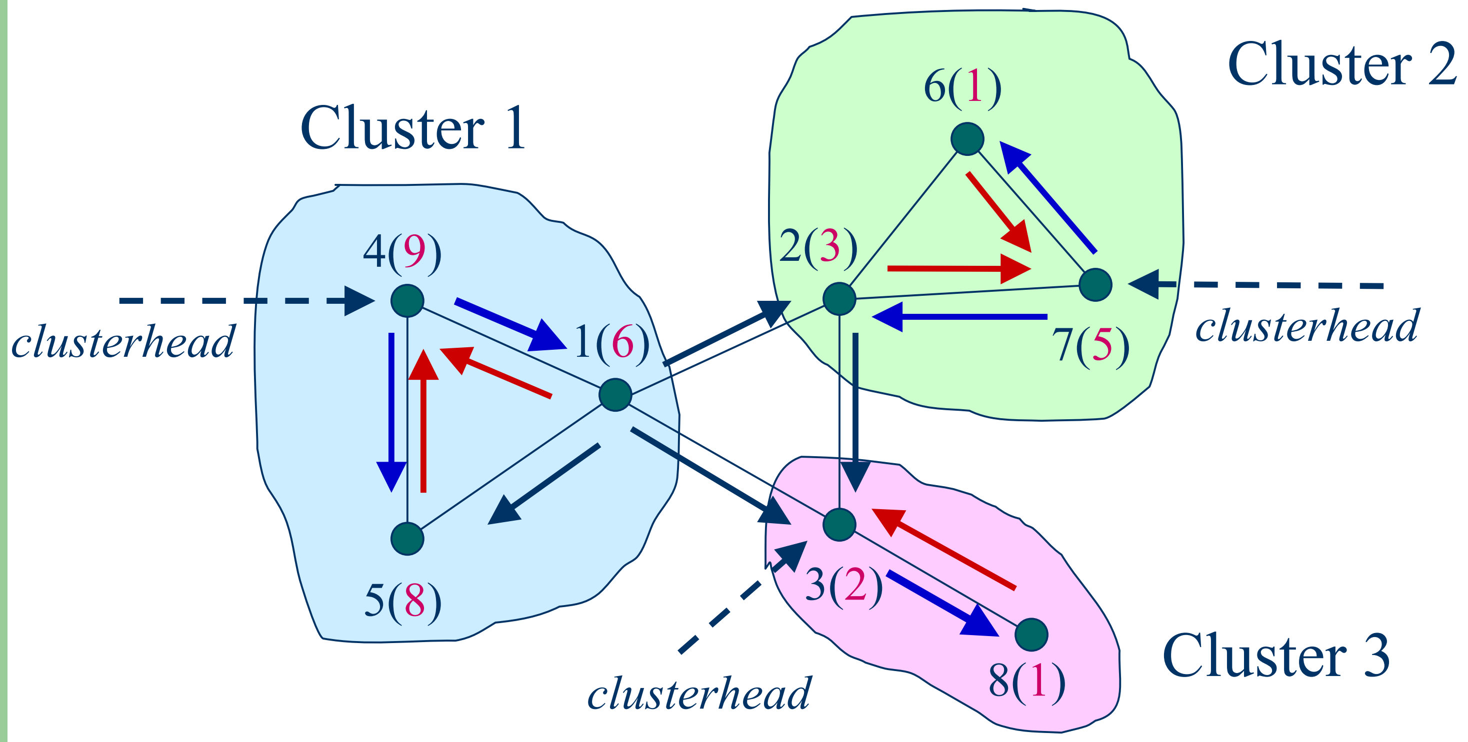
DCA: Distributed Clustering Algorithm

- Assumptions
  - Knowledge of IDs and weights of one-hop neighbors
  - Broadcast transmission of a packet in finite time (a "step")
  - Nodes do not move during clustering

DCA-Cont.

- (Only) Two messages:
  - CH(v): Sent by a clusterhead v
  - JOIN(u,t): Sent by ordinary node u when it joins the cluster of clusterhead t
- Three (simple) procedures:
  - Init (start up)
  - OnReceivingCH(v), OnReceivingJOIN(u,v) (message triggered)

Example



I Step II Step III Step IV Step V Step

DCA: Provable Properties

- Consider

$$\tau: V \rightarrow \{1,2,3, \ldots , 2k\}$$

V = set of network nodes, k = number of clusters

- **Proposition**: Each node v in V sends exactly one message by $\tau(v)$ steps
- **Corollary 1**: DCA message complexity is $n = |V|$
- **Corollary 2**: DCA terminates correctly in at most 2k steps ($\le 2n$)

A Note on the Average Time Complexity

- We notice that

$$k <= \alpha(G)$$

G = topology graph, $\alpha(G)$ = G's *stability number*

- We see the network as a *random graph*, for which

$$(2k <=)\ 2\ \alpha(G) = \text{circa } O(\log n)$$

Log's base is a function of n and the number of the network links

Adapting to Mobility and Node/Link Failures: DMAC

- DMAC is for clustering set up AND maintenance
- Nodes can move during the clustering
- Each node reacts to
  - Reception of a message
  - Presence of a new link
  - Link failure
- Same assumptions of DCA, plus knowledge of neighbors' roles (no role = ordinary role)

DMAC: The Procedures

- INIT
- Link-dependent procedures:
  – Link_Failure
  – New_Link
- Message-triggered procedures:
  – OnReceivingCH(v)
  – OnReceivingJOIN(u,t)

Joining Clusterheads: Dynamic Backbone

- A theorem from Chlamtac and Farago:

  If a network is connected, and DCA is used, then if and only if each clusterhead is linked to all the clusterheads at most three hops away, the resulting backbone network is connected
- Inherently mobility adaptive and stateless
- Good if the random graph model could be used

Dynamic Backbone: Some Simulation Results

- Networks with up to 2000 nodes (common parameters)
  - Number of Clusterheads < SQR(log n)
  - Number of Backbone Links < EXP(log n, 2.5) (when mapped on physical paths with at most three hops)
  - Number of Backbone links < EXP(log n, 2.2) (when links are obtained through directional antennas and/or power control)

Some Applications

- Stateless multipoint communication: Routing, multicast and broadcast over the backbone
- Resource/user discovery
- Implementation of security in large networks of sensors
- Network management

“To Go”

- DCA+DMAC for clustering ad hoc networks:
  - Clusterhead selection based on mobility/node, dynamically changing parameters
  - Executed at each node with minimal topology knowledge
  - DCA time complexity is bound to a network parameter that depends on the network topology
  - DMAC is mobility/failure adaptive, fast and easy to implement