Ad hoc Networking with Swarm Intelligence

Chien-Chung Shen

cshen@cis.udel.edu

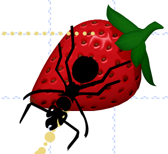
Outline

- Overview of swarm intelligence
- Multicast routing
- Topology control
- Conclusion

Overview of Swarm Intelligence

- "Complex" behaviors that arise from very simple individual behaviors and interactions
- Often observed in nature, especially among social insects such as **ants**
- Each individual has little intelligence, follows simple rules, and uses local information
  - Pheromone laying and following
  - Indirect communication – *stigmergy*
- (globally) optimized behaviors **emerge** when they work collectively as a group
- Example - food foraging with shortest path

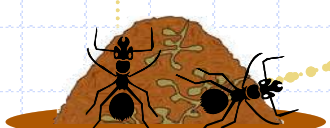
Swarm Intelligence



- Ants likely choose paths with higher pheromone intensity
- Trail gets reinforced ***(positive feedback)***

Without reinforcement, pheromone evaporates ***(negative feedback)***

Ants lay pheromone

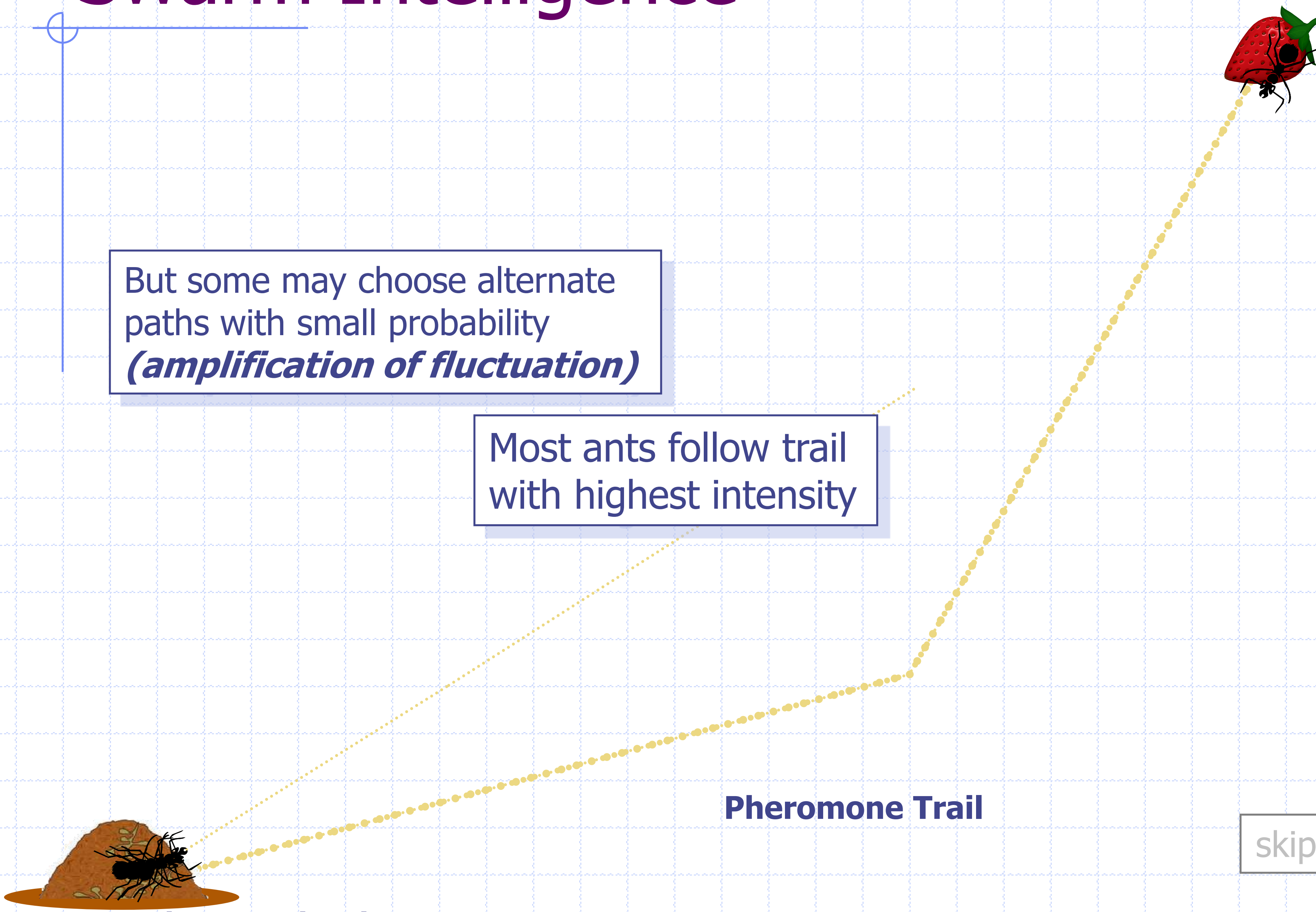


skip

Swarm Intelligence

But some may choose alternate paths with small probability ***(amplification of fluctuation)***

Most ants follow trail with highest intensity



skip

Essence of Swarm Intelligence

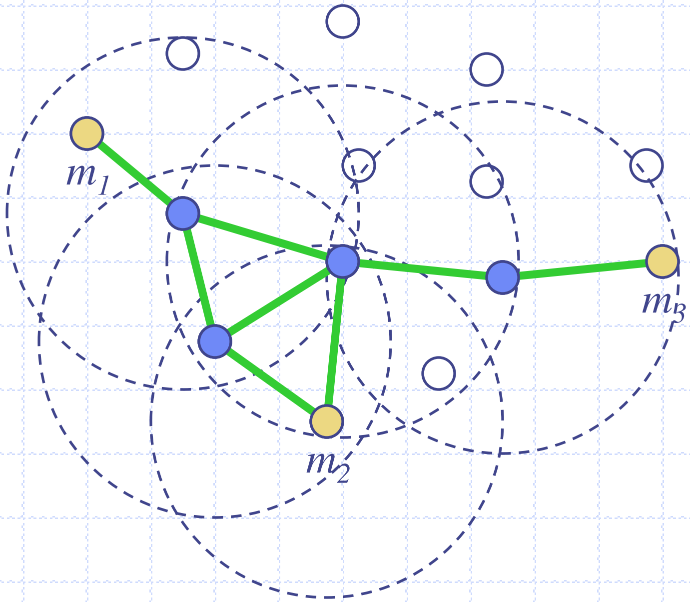
- Positive and negative feedback
  - search good solutions and stabilize the results
- Amplification of fluctuation
  - discover new solutions and adapt to changing environment
- Multiple interactions
  - Allows collaborations among distributed entities to coordinate and self-organize
- **A distributed adaptive control system**

Ad hoc **N**etworking with **SI (ANSI)**

- Unicast routing (ANSI)
- **Multicast routing (MANSI)**
- **Topology control (ABTC)**
- Energy conservation (ABEC)
- Feature interactions – cross-layer and cross-feature

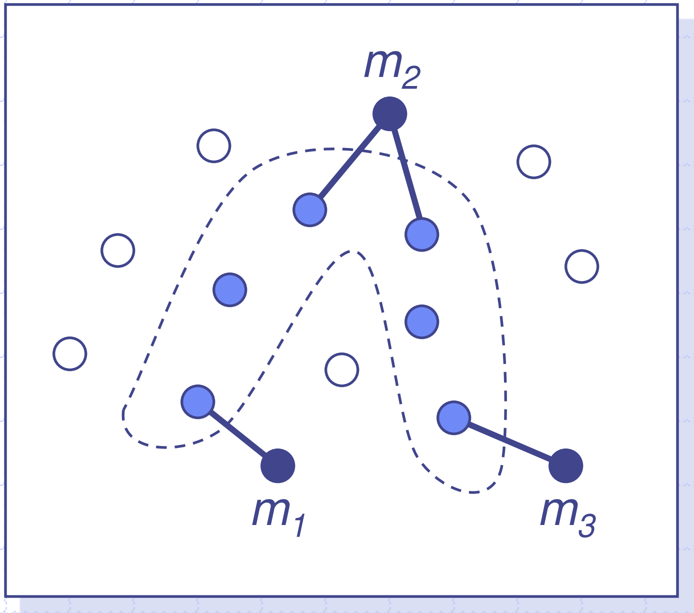
MANSI: Overview

- A subset of nodes, called a **forwarding set**, are extracted to rebroadcast data packets
- The forwarding set is *shared* among all group members (**group-shared approach**)
- Each forwarding node always rebroadcasts non-duplicate packets, regardless of the previous hop (**mesh-based approach**)
- The forwarding set is constructed only when some sources have data to send (**reactive approach**)

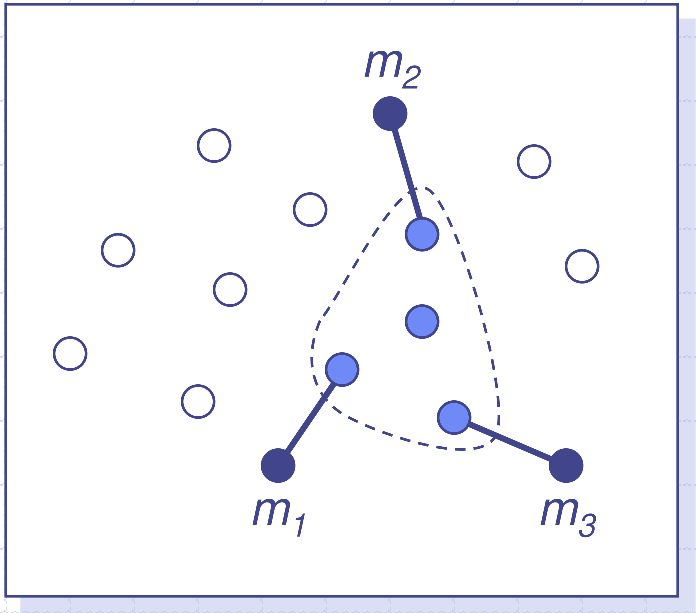


MANSI: Overview

- Multicast connectivity is more efficient when group members **share** existing forwarding nodes



Forwarding set of **6** nodes



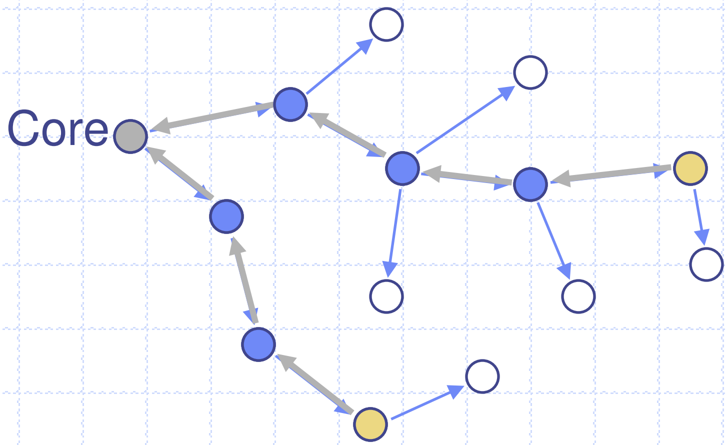
Forwarding set of **4** nodes

Protocol Operations

- Two phases:
  - Forwarding Set **Initialization**
    - A forwarding set is rapidly constructed on-demand
    - Efficiency of the forwarding set is not the main concern at this phase
  - Forwarding Set **Evolution**
    - Ant packets are deployed to explore and discover better forwarding sets

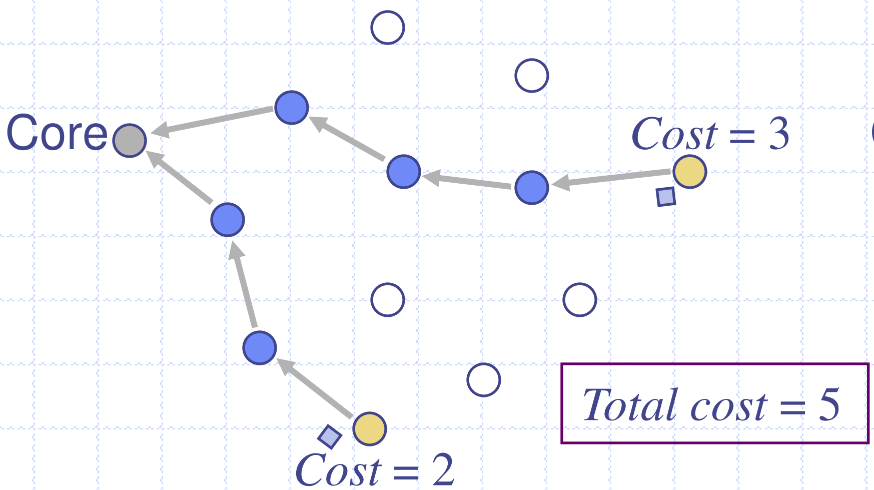
MANSI: Protocol Operations

- Forwarding set initialization
  - The first sender of the group becomes the *core* node, and floods an announcement
  - Other members request to join the group via the reverse paths

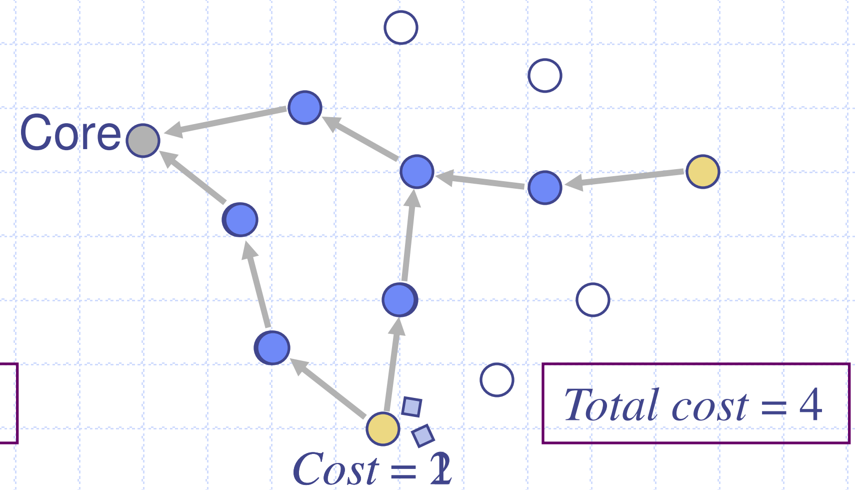


MANSI: Protocol Operations

- Forwarding set **evolution**
  - **Forward Ants** (**packets**) are deployed by members to *opportunistically* discover new connectivity that yields lower *cost*
  - A Forward Ant turns into a **Backward Ant** when it encounters another existing path and returns to its originator



Ants follow current best paths and update costs

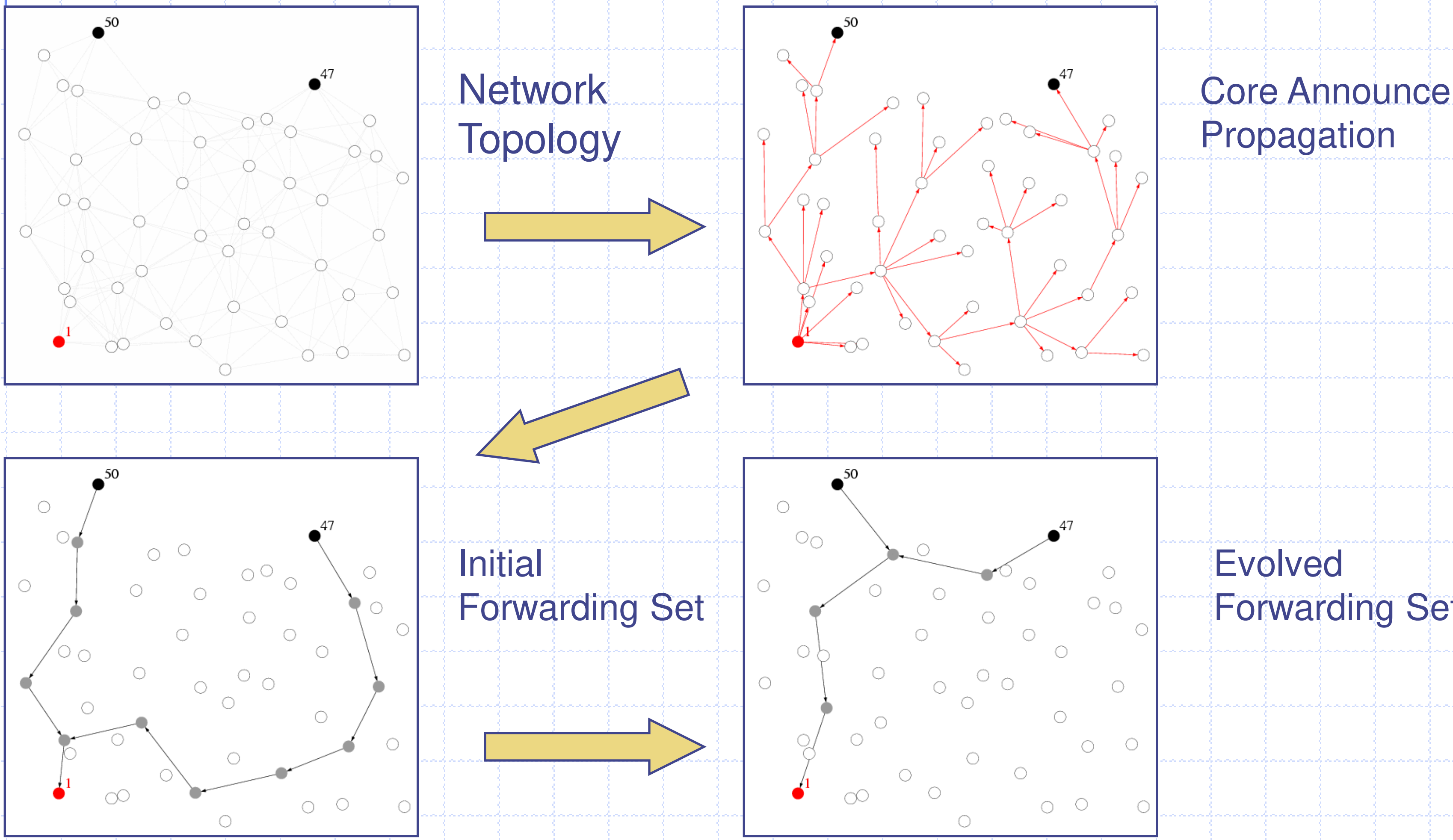


Ants **opportunistically** discover other paths

MANSI: Protocol Operations

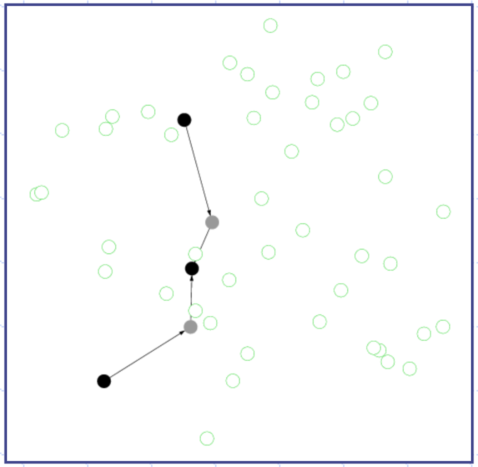
- A **Backward Ant** deposits pheromone on the returning trip, where the amount of pheromone is *inversely* proportional to the cost of the trip
  - → **The shorter the trip, the higher pheromone amount it deposits**
- Each member selects the next hop with the highest pheromone intensity to connect to the core
- To prevent two members from connecting to each other's path (a race condition), a Forward Ant from a member *m* is allowed to turn back only when it encounters a forwarding node used by another member m', where $m' > m$
  - Member with the highest ID can only connect to the core
  - Member with the lowest ID can connect to any existing forwarding node, except one of its own

MANSI: Sample Snapshots

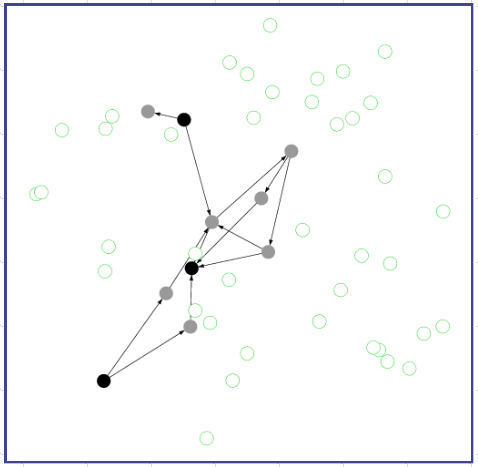


MANSI: Adapting to Mobility

- With mobility, multicast connectivity becomes fragile
- MANSI with mobility-adaptive mechanism
  - Each node keeps track of *link failure frequency* which indicates **stability** of its surrounding area.
  - When link failure frequency is higher than a threshold, a forwarding/member node picks **two** forwarding nodes with highest pheromone intensities, instead of one



Without mobility-adaptive



With mobility-adaptive –
more robust group connectivity

MANSI: Simulation

- QualNet simulator
- Simulation setup

Terrain dimension	$1000 \times 1000\ m^2$
Communication range	250 m
Mobility speed	0-20 m/s
# Nodes	50
# Members/# Senders	5
Application Traffic	CBR (1 KB/s from each sender)
Core announce interval	10 seconds
Ant deploying interval	2 seconds

MANSI: Simulation Results

Size of forwarding set in static network

Network	Average Size		
	MANSI	CORE	FLOOD
1	7.89	9.49	50.00
2	4.00	3.67	50.00
3	4.00	4.97	50.00
4	4.46	4.68	50.00
5	6.51	8.46	50.00
6	5.52	6.25	50.00
7	6.90	7.83	50.00
8	6.04	7.46	50.00
9	5.16	7.67	50.00
10	5.02	6.95	50.00

MANSI → Forwarding set evolves by deploying ants

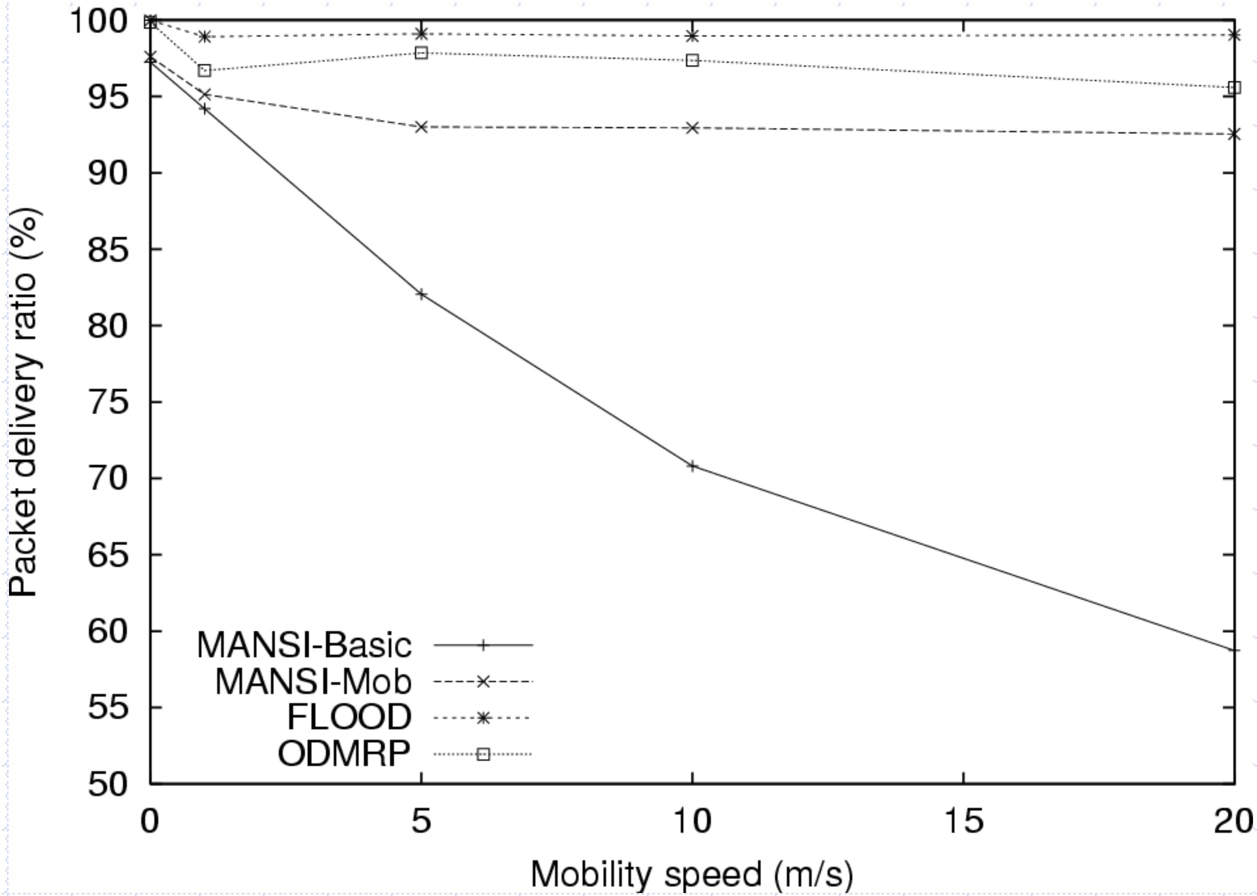
CORE → Forwarding set is constructed only by core announcement (no ants deployed)

FLOOD → Every node is in the forwarding set (flooding)

Forwarding sets discovered by ants are 20% smaller

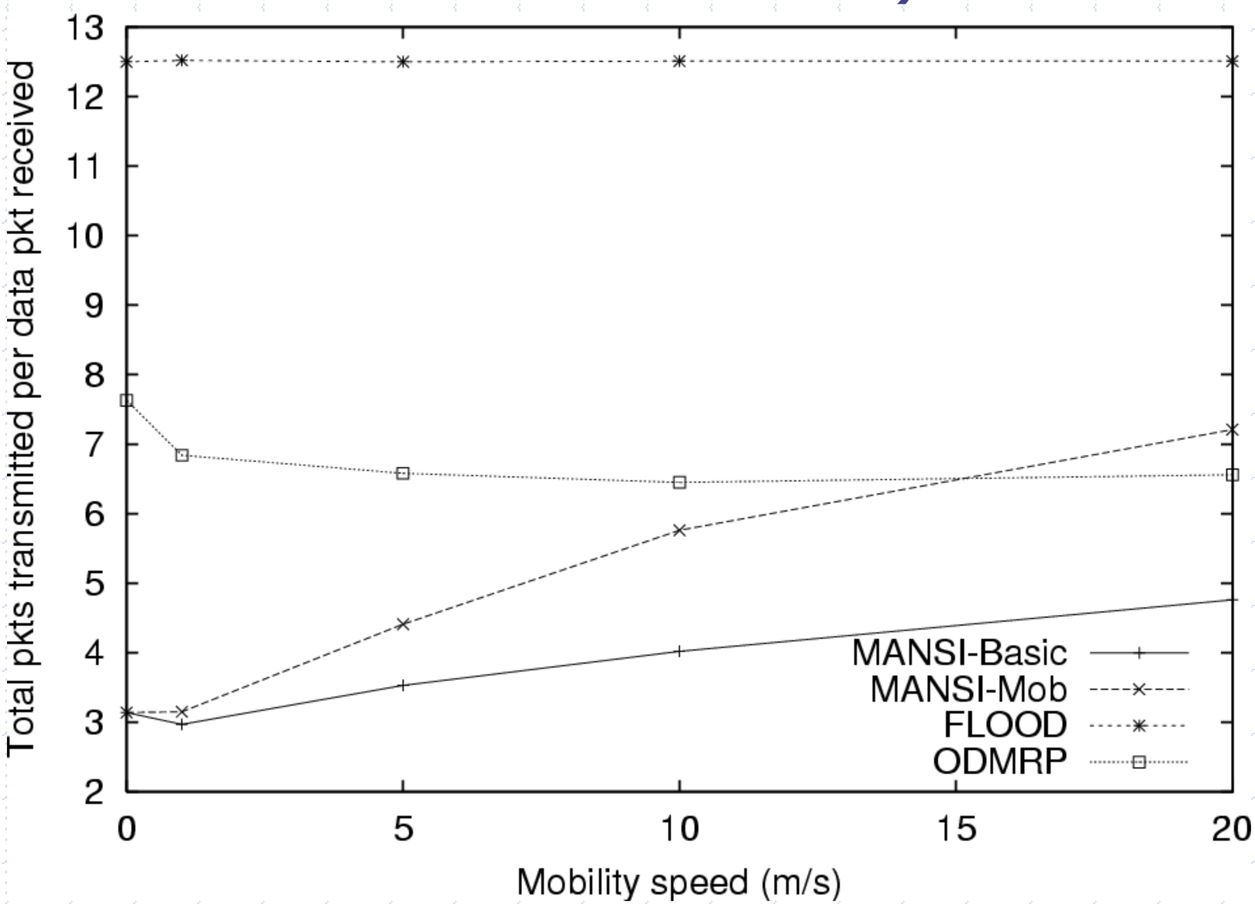
MANSI: Simulation Results

- Packet delivery ratio (**effectiveness**)



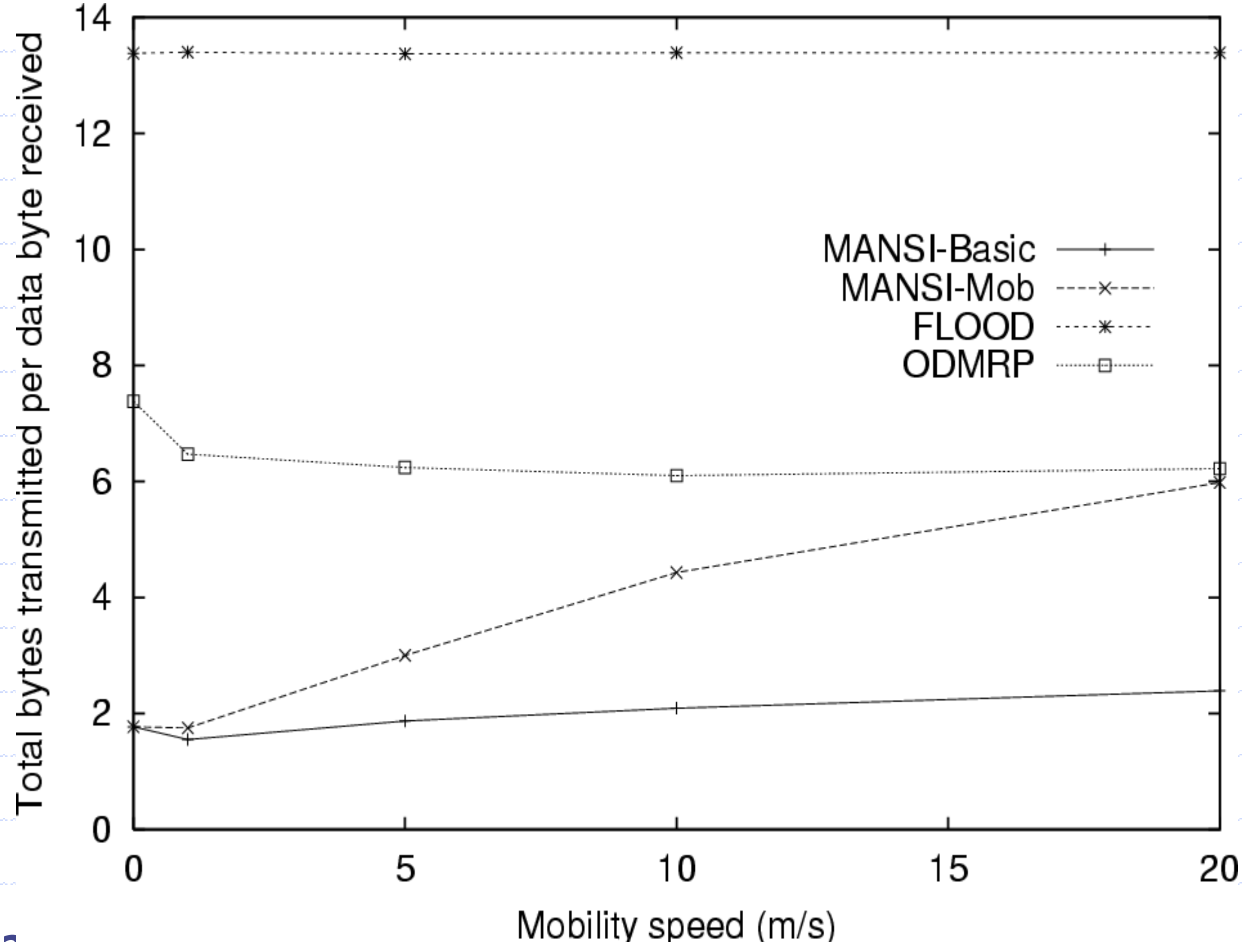
MANSI: Simulation Results

- Total **# of packets** (HELLO, Ant, Data) transmitted per data packet received (**efficiency in terms of channel access**)



MANSI: Simulation Results

- Total **bytes** (HELLO, Ant, Data) transmitted per data byte received (**efficiency in terms of bandwidth utilization)**

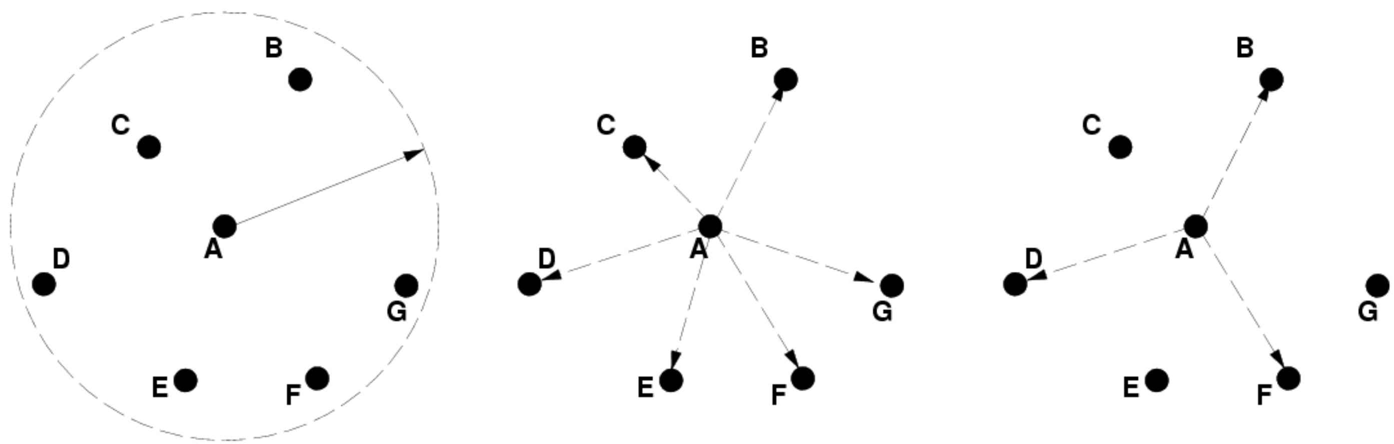


MANSI: Summary and Future Work

- Ant packets are deployed to **opportunistically** discover new paths resulting in reduced total cost (*e.g.*, number of nodes) of the forwarding set
- With different cost functions, MANSI is being applied to:
  - Reliable multicast
  - Load balancing
  - Energy-aware routing
  - Energy conservation
  - Secure routing

Topology Control

- Objectives
  - Reduce transmission power and conserve energy
  - Reduce interference and increase effective network capacity
- Approaches
  - Physical topology control – *e.g.* **ABTC**
    - Power adjustment
  - Logical topology control – *e.g.* CBTC
    - Power adjustment plus **neighbor selection**



(a) Node coverage with transmission power (b) Physical topology (c) Logical topology

Ant-Based Topology Control (ABTC)

- Overview
  - Problem formulation
  - Features
- Operations
- Simulation
- Animations

ABTC: Overview

- Problem formulation
  - "minimize" **maximum** (**MinMax**) or **total** (**MinTotal**) power while maintaining connectivity
- Basic idea
  - Every node periodically broadcasts ant packets with various power levels to be forwarded by its full-power neighbors
  - Upon receiving an ant packet, a node makes a decision on whether to forward the packet, updates its local information using the packet, re-evaluates a local condition based on such information to assign a proper power level
- Features
  - **No** AOA, GPS, routing, or topology information used
  - **Distributed, asynchronous,** and **localized**
  - Inherently **adaptive to mobility**

ABTC

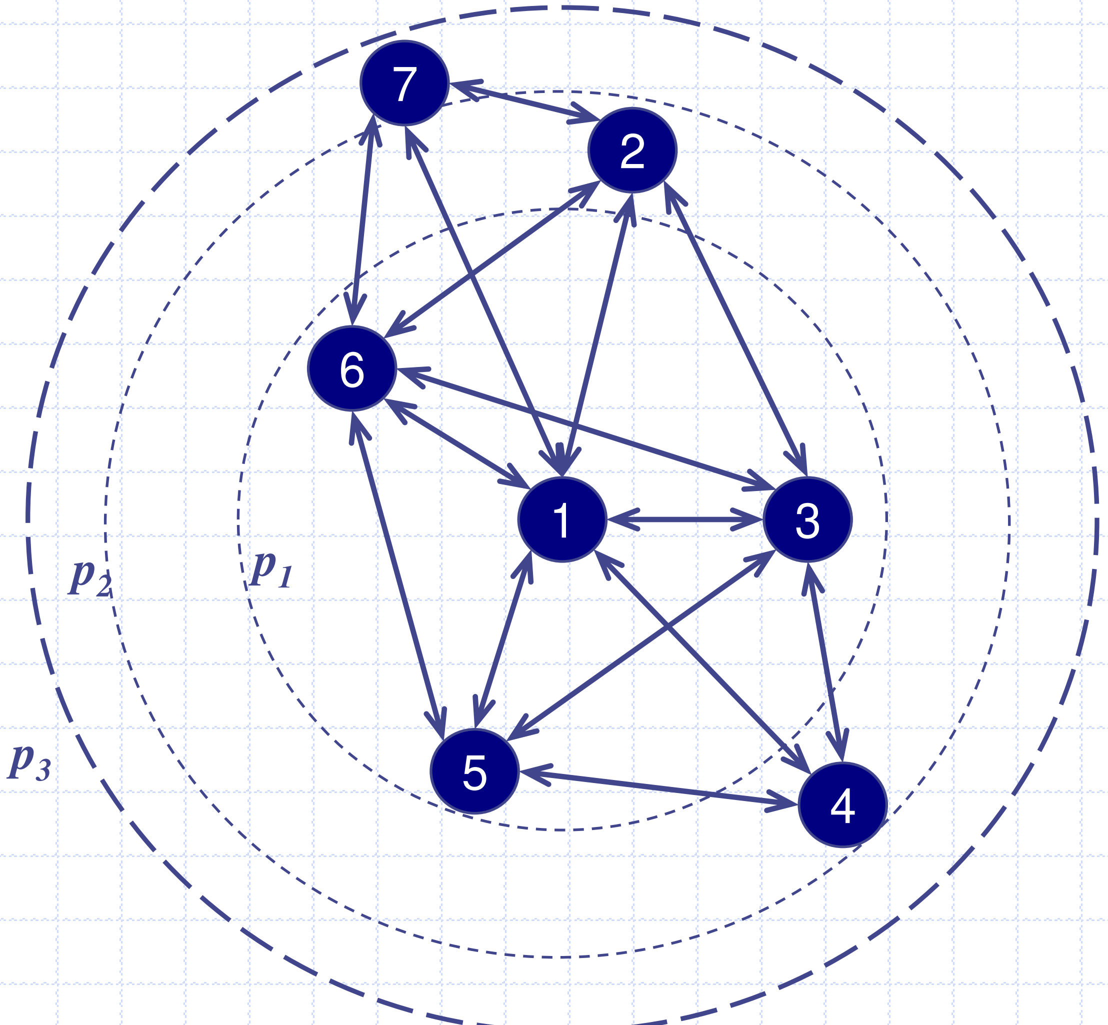
- Overview
- Operations
  - Maintain connectivity
  - Adapt to mobility
  - Minimization objectives
  - Interworking with routing
- Simulation
- Animations

Maintain Connectivity

- Periodic "neighbor" discovery via full-power
- Periodically broadcast ant (packet) using a certain power level (**P**), and the ant is relayed (re-broadcast) with **P** by neighbors
- Upon receiving ants, a node determines the (locally) **minimal** power ($\mathbf{P_{min}}$) such that it can receive ants **originated from *all*** its neighbors and **relayed *only* by** its neighbors
- **Rule**: If the condition is satisfied, the node is assigned $\mathbf{P_{min}}$
- Theorem: **If every node follows the rule, the resulting power assignment guarantees network connectivity**

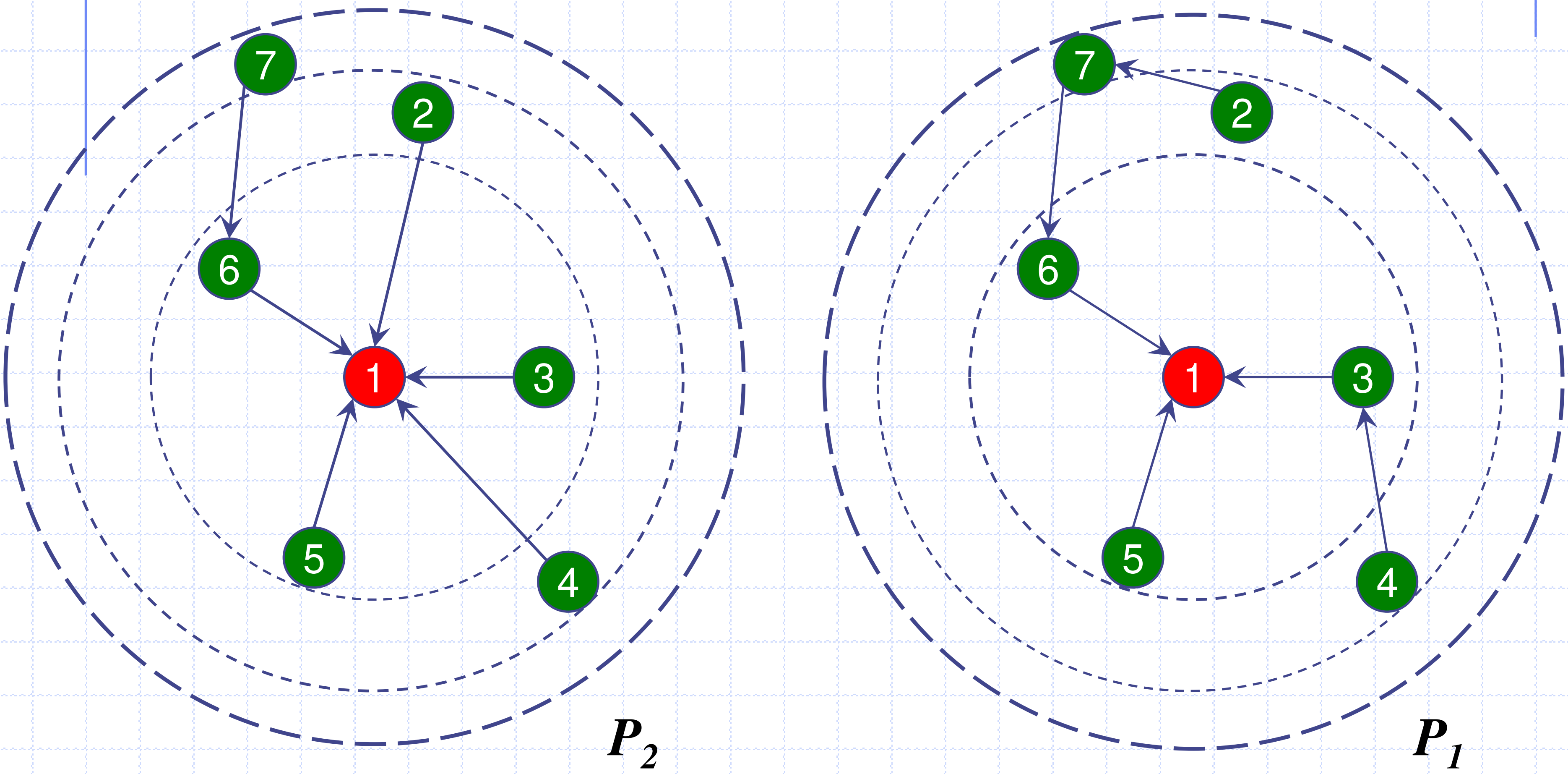
Example (1)

- Initial topology with full power (p_3)



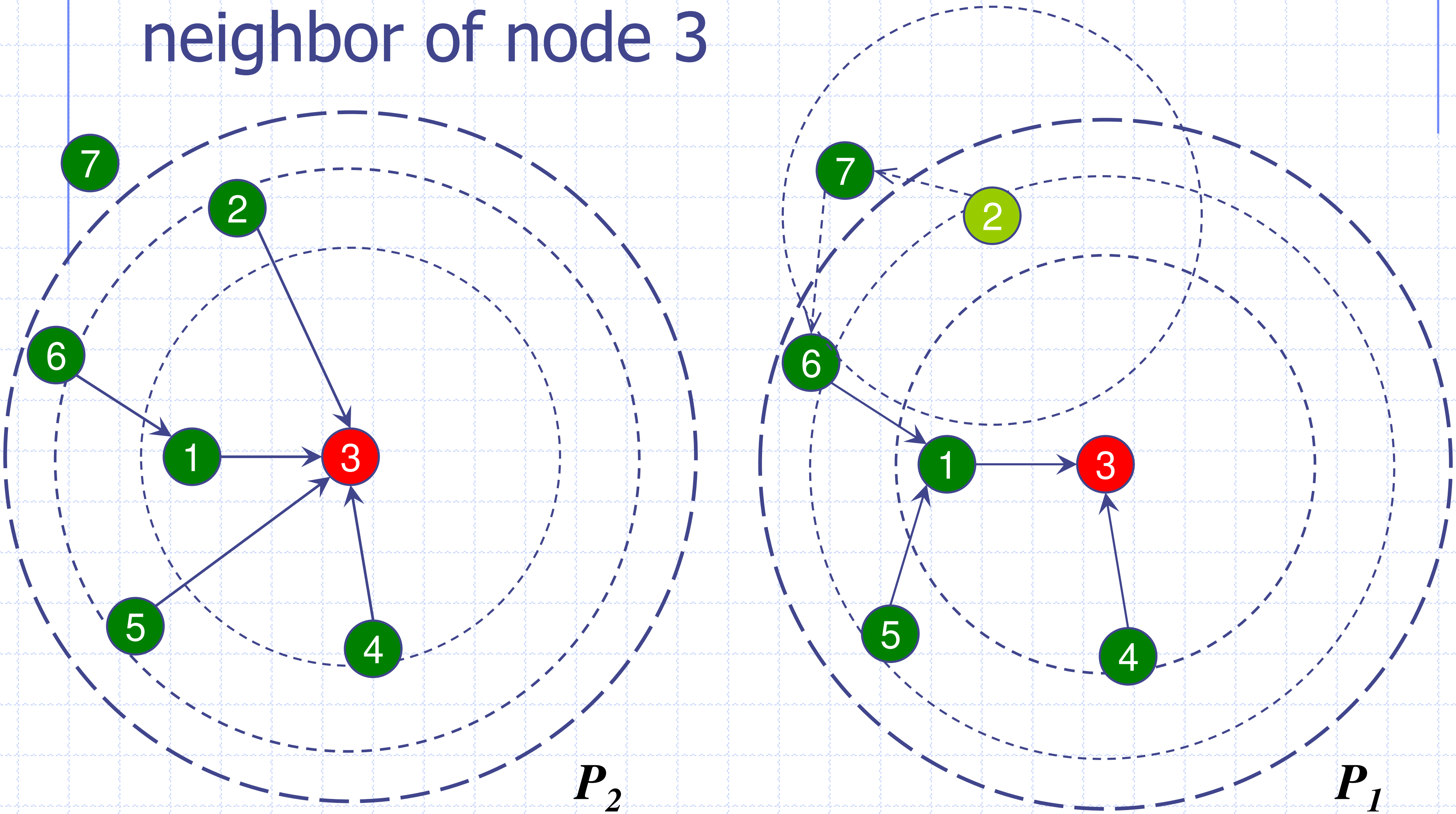
Example (2)

- Ants received at node 1 ***originated from all*** and ***relayed only by*** neighbors



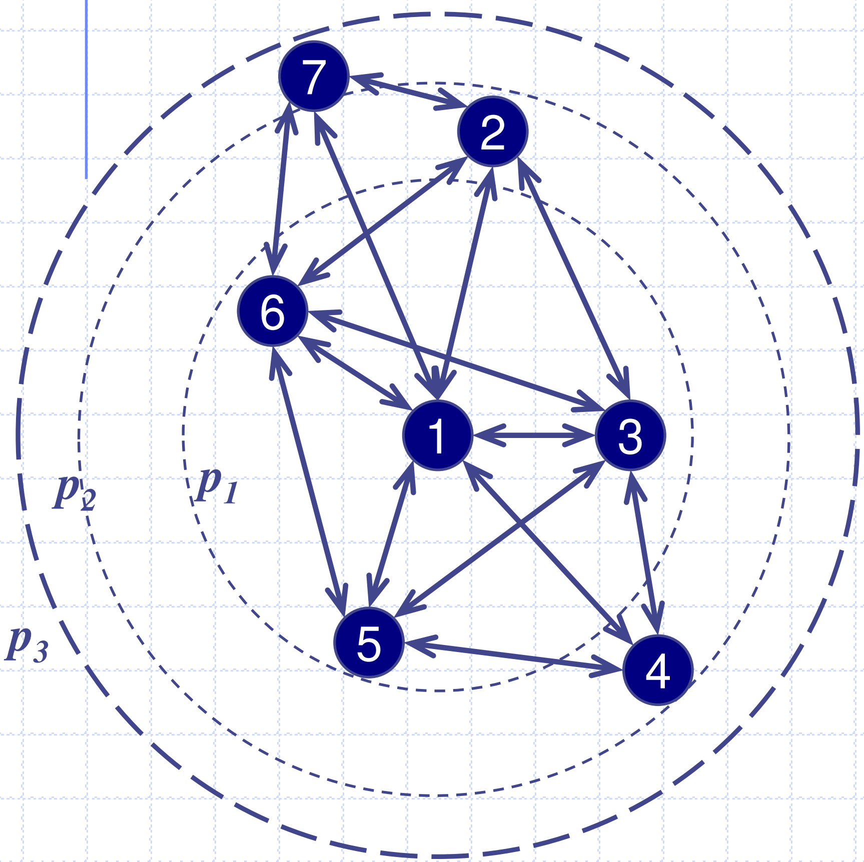
Example (3)

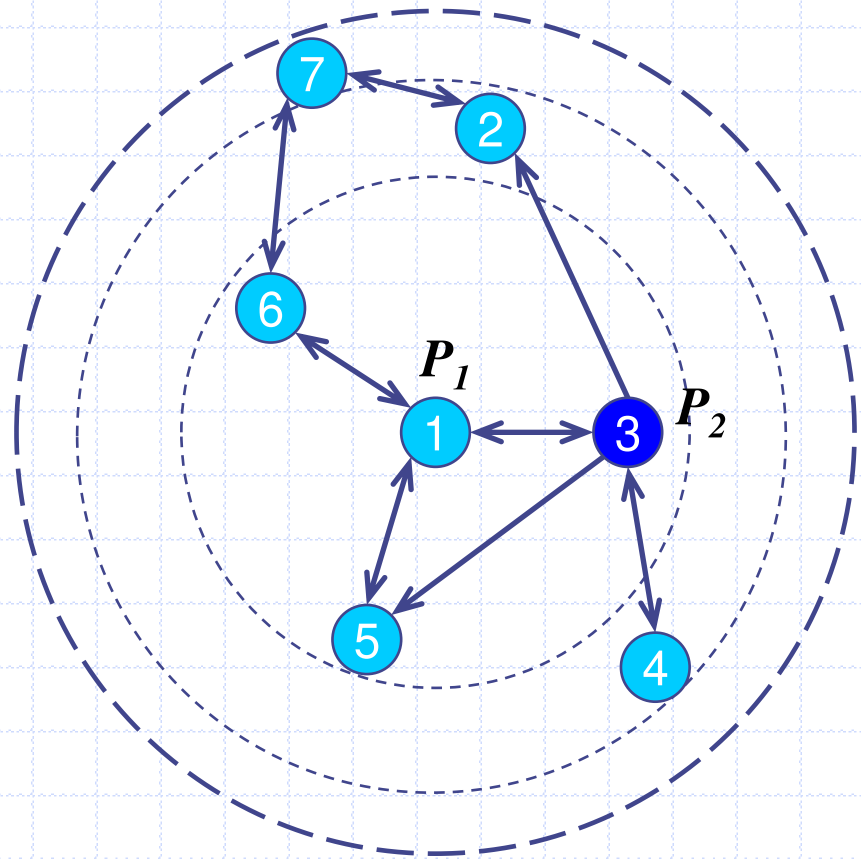
- Ants received at node 3 – node 7 is not a neighbor of node 3



Example (4)

- Initial topology ➔ Final topology





Adapt to Mobility

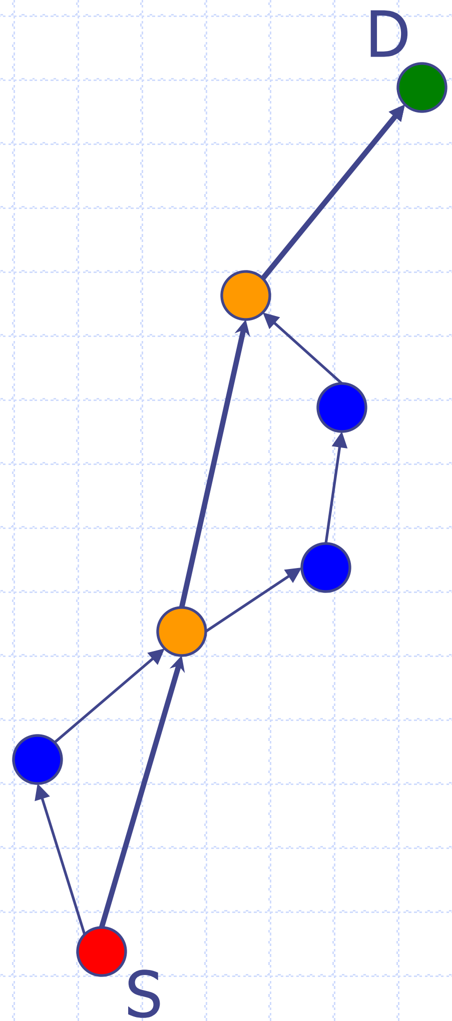
- Without mobility, every node originates ants using each power level once, and collects ants from all neighbors to determine power assignment → a **search problem**
- With mobility, need ***heuristic*** search to **converge faster**
- Swarm intelligence
  - **Positive feedback** to expedite the convergence to good solutions
  - **Amplification of fluctuation** to opportunistically discover better solutions
- Pheromone: **goodness** of power assignment
  - Goodness is proportional to the **inverse** of power level
  - Every node maintains goodness of power assignment in a "**pheromone table**" to generate heuristics for optimal local solutions in which a node receives ants with the **lowest common power** from **maximum number of neighbors**
  - The values in the pheromone table are updated by ants and degrade with time (**pheromone evaporation**)

Minimization Objectives

- **MinMax – minimize the maximum power used by any node in the network**
  - Node uses its current power assignment *with higher probability* as the power level to originate ants
  - Once a node converges to a power level, more ants will be originated with this power level, which in turn reinforce its neighbors' pheromone value associated with this power level → **COMPOW** [Kumar]
- **MinTotal – minimize the total power used by all of the nodes in the network**
  - Node chooses all available power levels with equal probability to originate ants
  - The pheromone value of a node is reinforced by neighbors → **ClusterPow** [Kumar]

Interworking with Routing

- ABTC executes **independently** from the routing protocol
- ABTC provides a way of forwarding data packets without affecting routing
- Instead of forwarding data packets to the next full-power neighbor, ABTC forwards them **via multi-hops within the neighborhood with assigned power**
- The route information within the neighborhood is collected from the received ant packets



ABTC

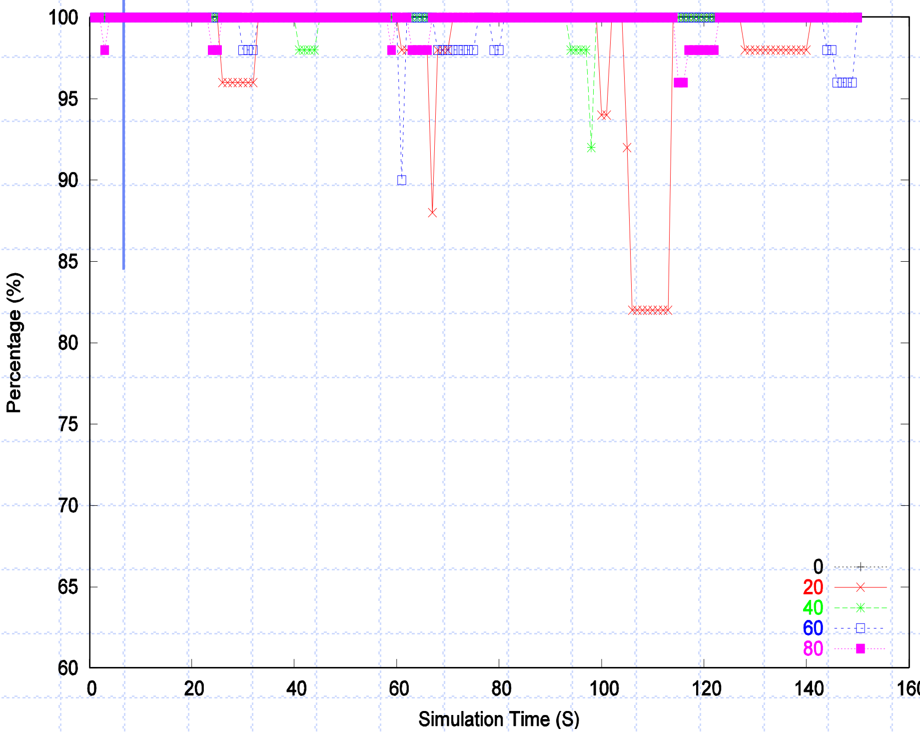
- Overview
- Rationale
- Simulation
  - Environment
  - Level of connectivity
  - Power (maximum/average)
  - Average end-to-end total power
  - Average end-to-end hop count
- Animations

Environment

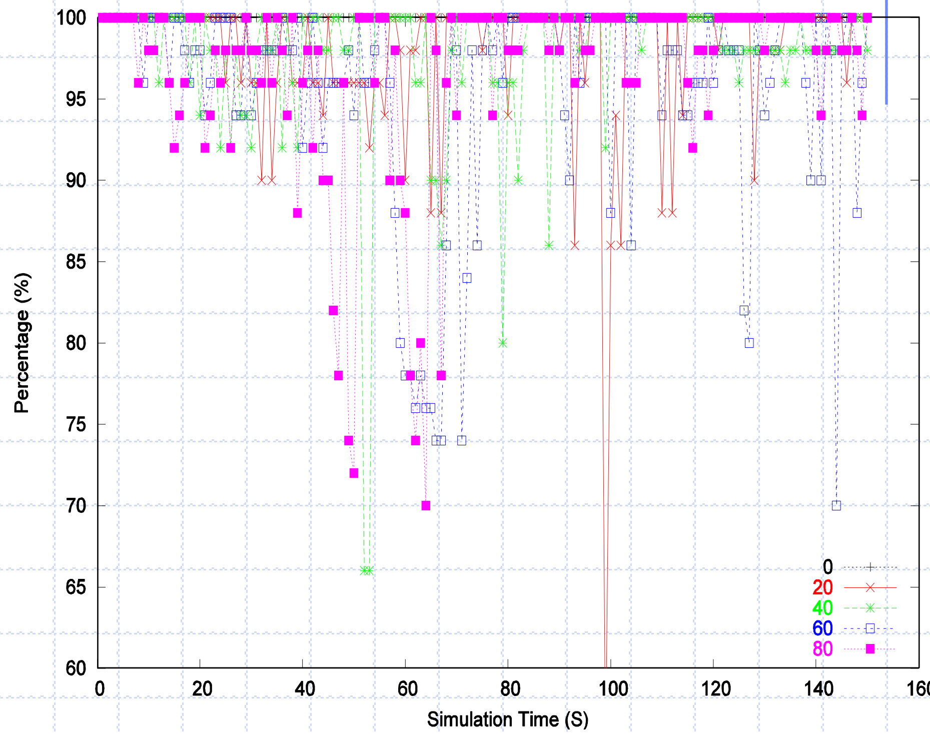
- QualNet simulator
- Modified IEEE 802.11 MAC to support programmable transmission power
- Terrain size: 1500 x 1200 m^2
- Number of nodes: 50
- 7 Power Levels: 2.58, 5.08, 7.89, 11.05, 13.73, 16.04, 18.10 dBm
- Power Ranges: 150, 200, 250, 300, 350, 400, 450 m
- Path-loss model: two-ray
- Raw data rate: 2 Mb/s
- Min/Max Speed: 10/[0,20,40,60,80] m/s
- **Comparison study**
  - CENT – centralized algorithm like CONNECT [Ramanathan]
  - CBTC – Cone-Based Topology Control [Wattenhofer,Halpern]

Level of Connectivity over Time

Measure how well the network is connected

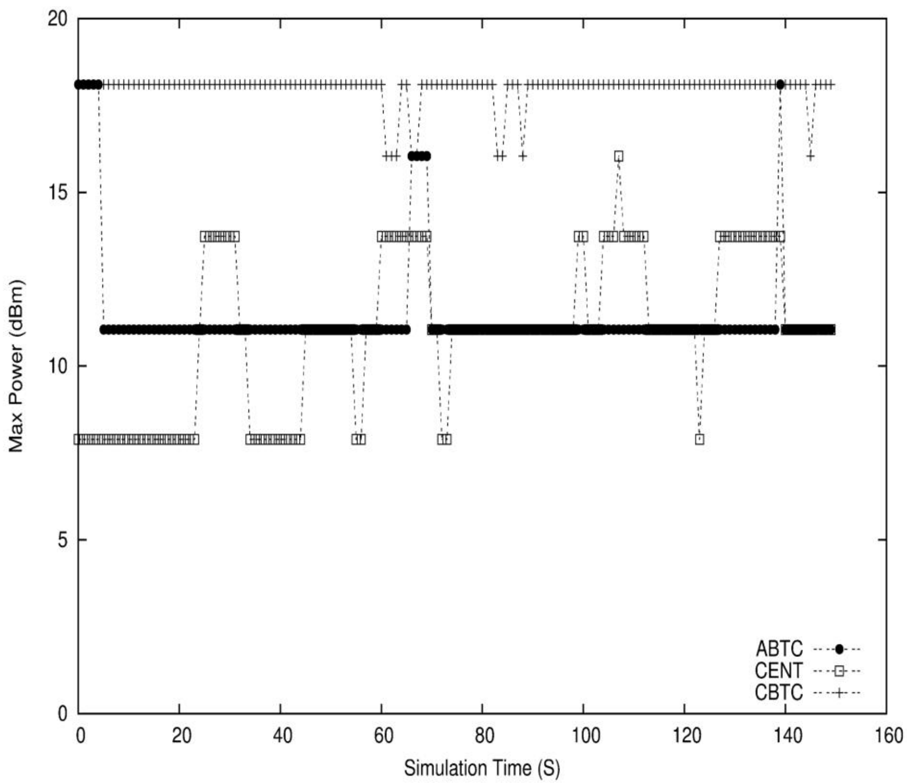


ABTC-MinMax

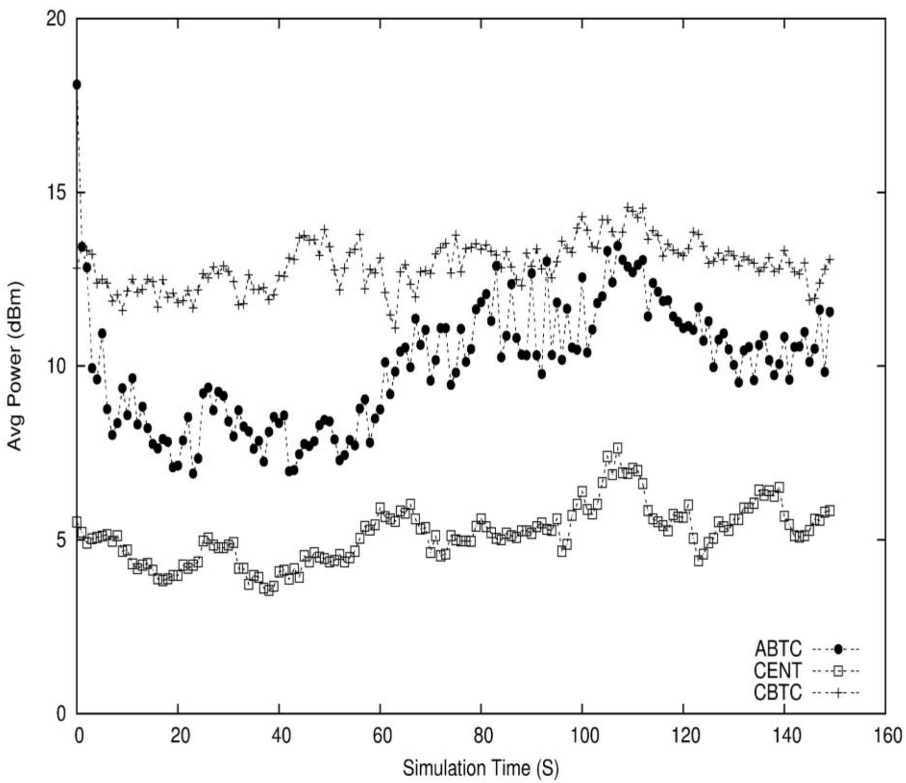


ABTC-MinTotal

Power (Maximum & Average)



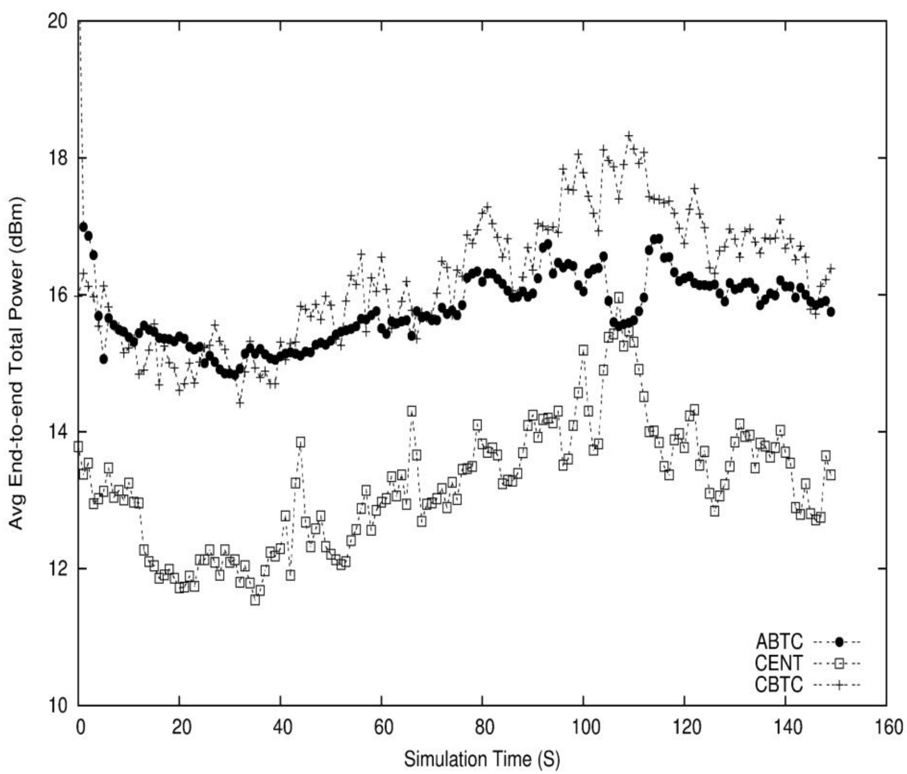
ABTC-MinMax
(max power)



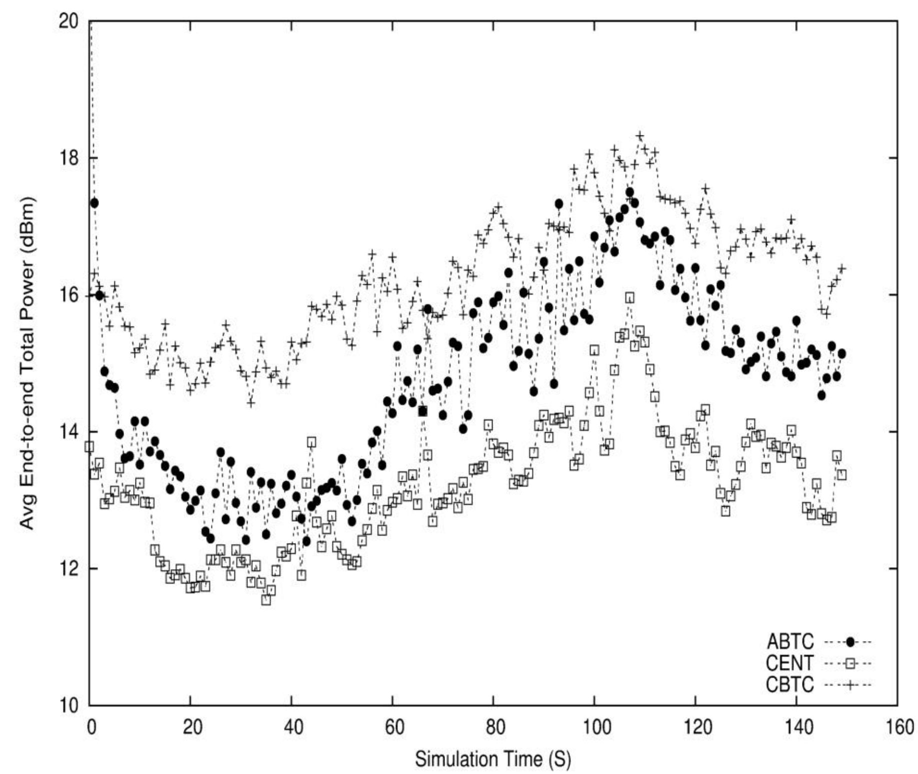
ABTC-MinTotal
(avg power)

Average End-to-End Total Power

Measurement of end-to-end energy consumption



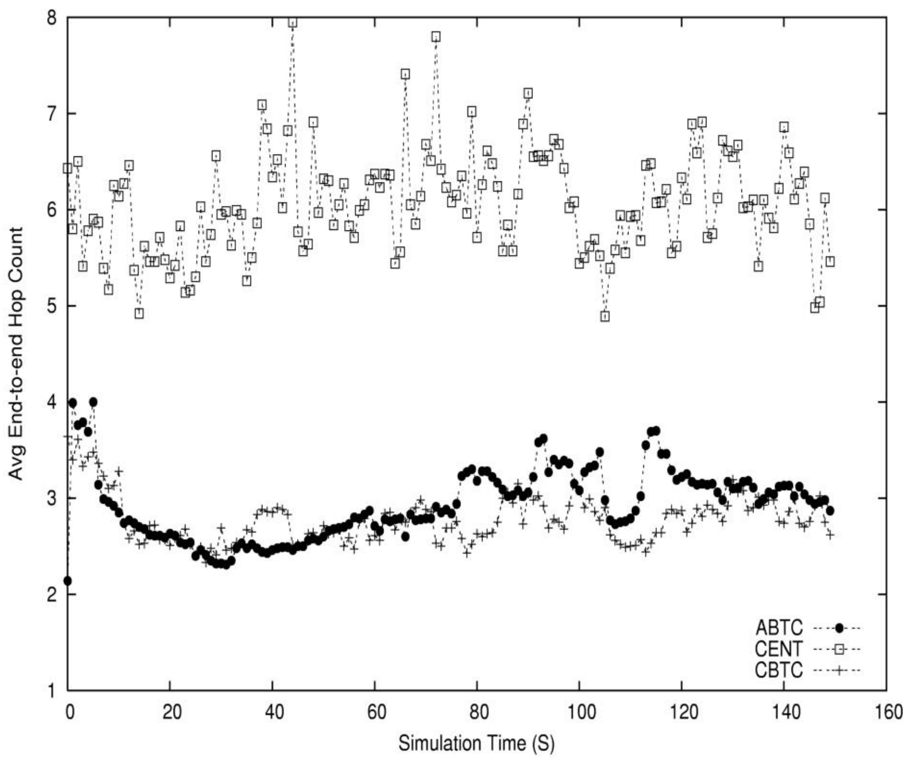
ABTC-MinMax



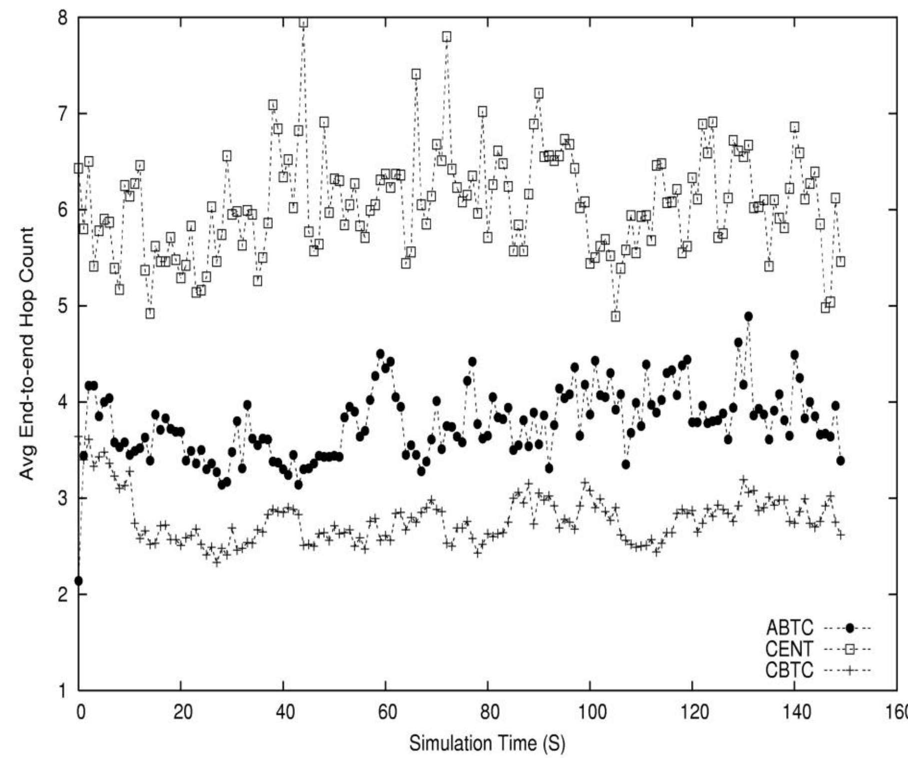
ABTC-MinTotal

Average End-to-End Hop Count

Measurement of end-to-end delay



ABTC-MinMax

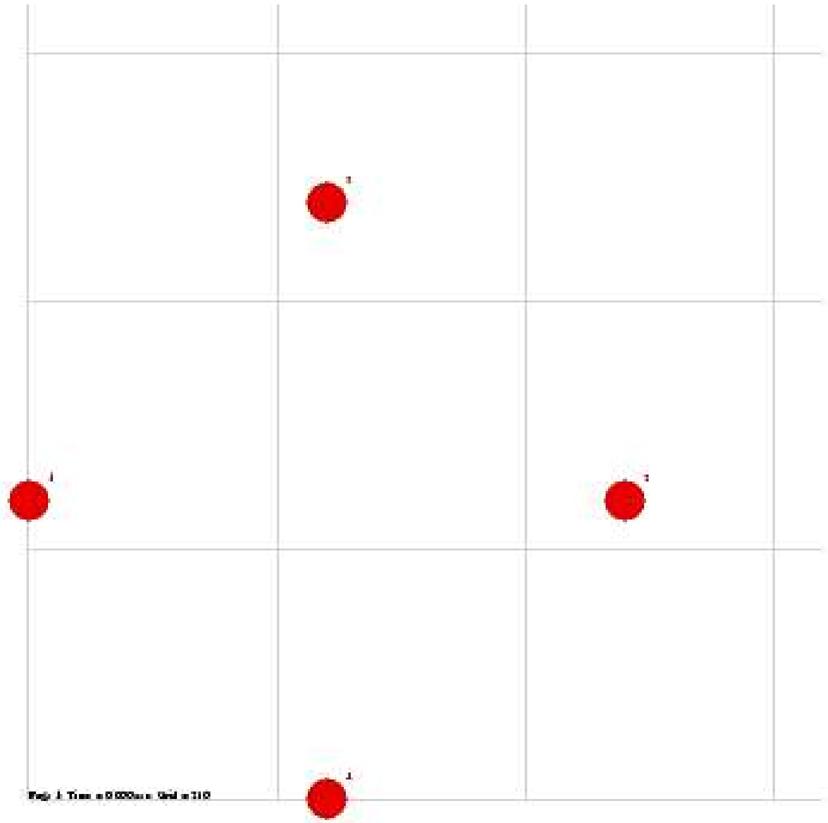


ABTC-MinTotal

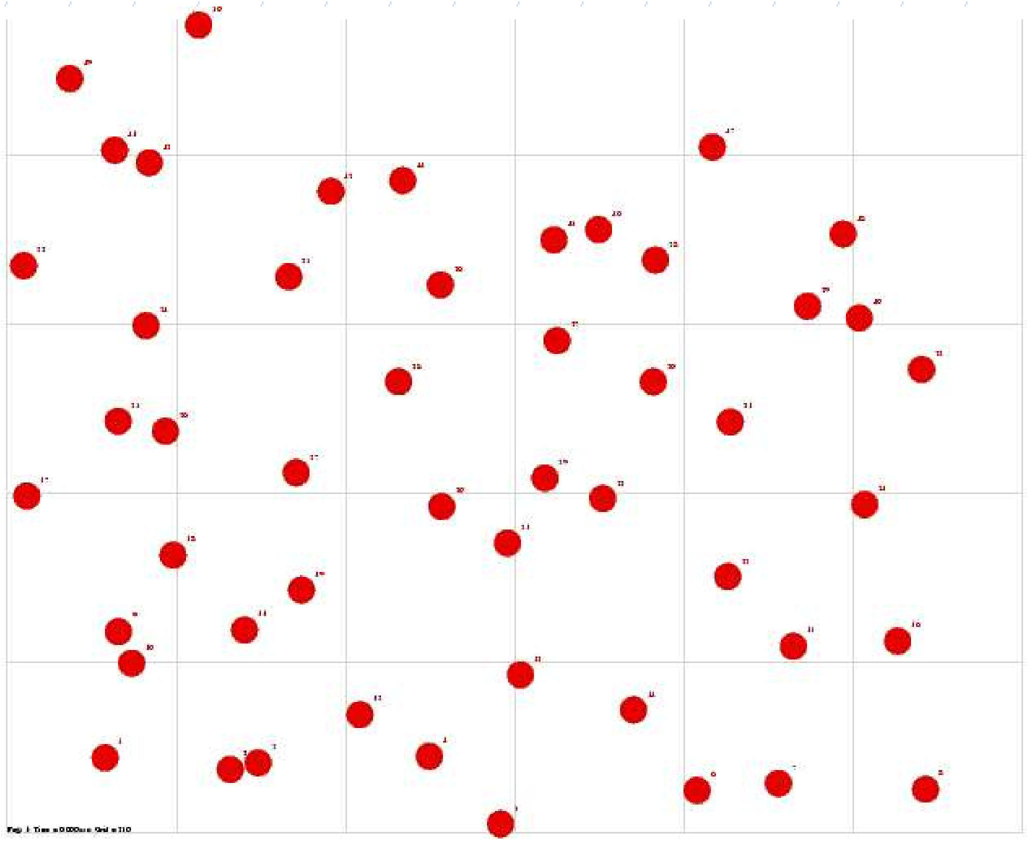
ABTC

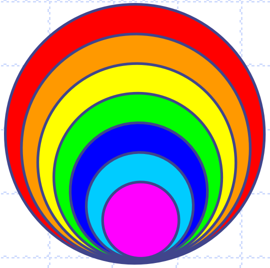
- Overview
- Rationale
- Simulation
- Animations

Animation (1)



Animation (2)





Conclusion

- **Swarm intelligence**
  - Positive and negative feedback
  - Amplification of fluctuation
  - Multiple interactions
- **A distributed adaptive control system**
- **Ad hoc Networking with Swarm Intelligence**
  - Unicast routing (ANSI)
  - Multicast routing (MANSI)
  - Topology Control (ABTC)
  - Energy conservation (ABEC)
  - Feature interactions – cross-layer and cross-feature