

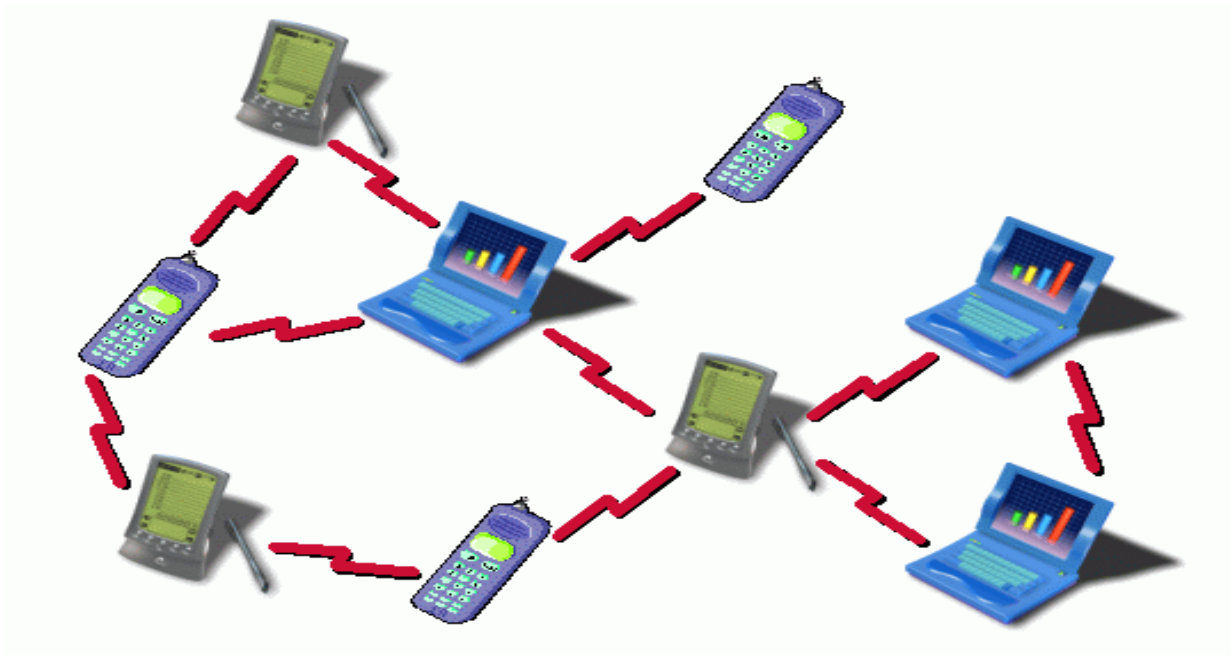
# Power Aware Management Middleware for Multiple Radio Interfaces

Roy Friedman, Alex Kogan

Computer Science, Technion

# Mobile ad hoc networks: what?

- Formed by wireless mobile devices
- No fixed infrastructure
- Nodes route messages over multiple hops
- Devices are battery-operated



# Mobile ad hoc networks: why?

- Fast and easy deployment
  - temporary social and professional events
    - conferences, exhibitions, ...
  - vehicular networks
  - emergency military operations
  - natural disasters
- Zero air-time cost
- High bandwidth and low latency

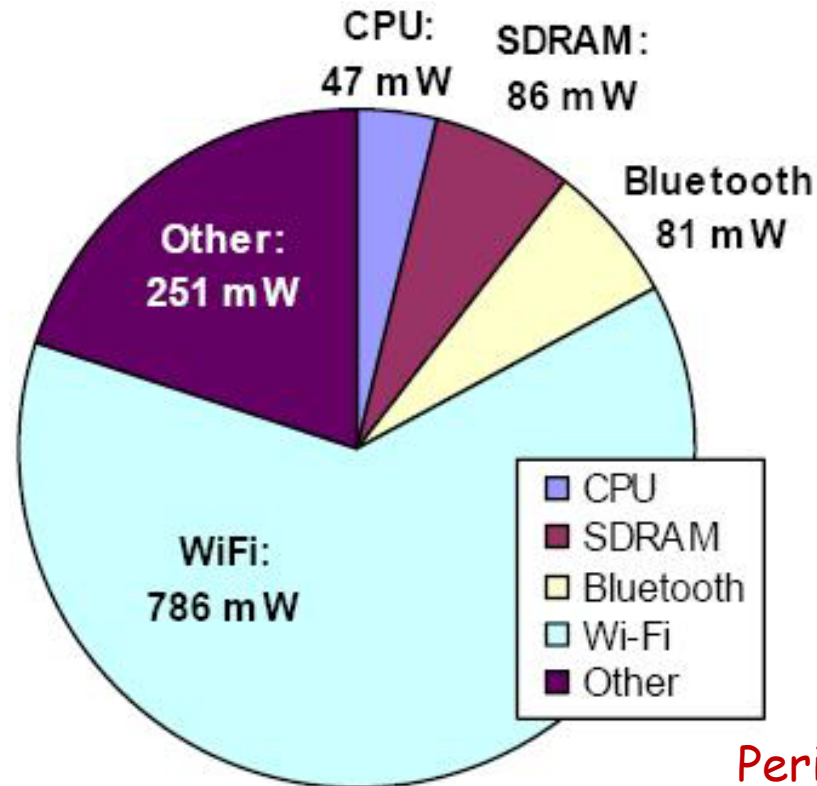
# Mobile ad hoc networks: how?

- Several standards for radio interfaces
  - well-established : WiFi, Bluetooth
  - emerging soon: ZigBee, WiMax, ...
- Modern devices are equipped with multiple interfaces



# Power consumption of a mobile device

- Wireless communication consumes most power



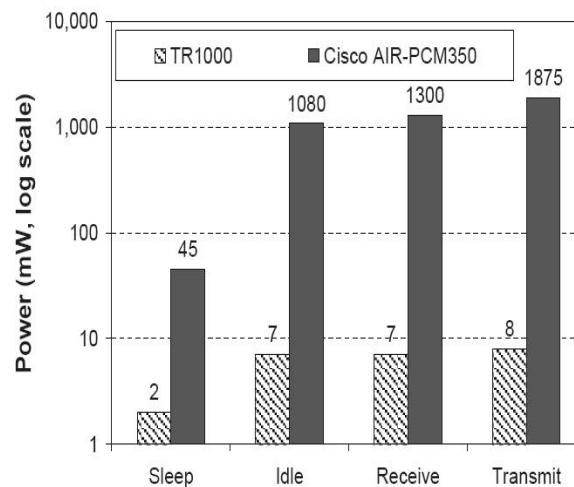
*Pering et. al. @ MobiSys '06*

Power breakdown for an idle connected mobile device.

The wireless interfaces consume approximately 70% of the total power.

# Multiple interfaces - a brief comparison

- Radio technologies differ vastly in:
  - transmission range
    - e.g., 10m for BT, 100m for WiFi
  - energy requirements
    - e.g., <100mW for BT, >1000mW for WiFi
  - bandwidth
    - e.g., 1Mbps for BT, 11/54Mbps for WiFi
- Idle power is not negligible



# Previous work

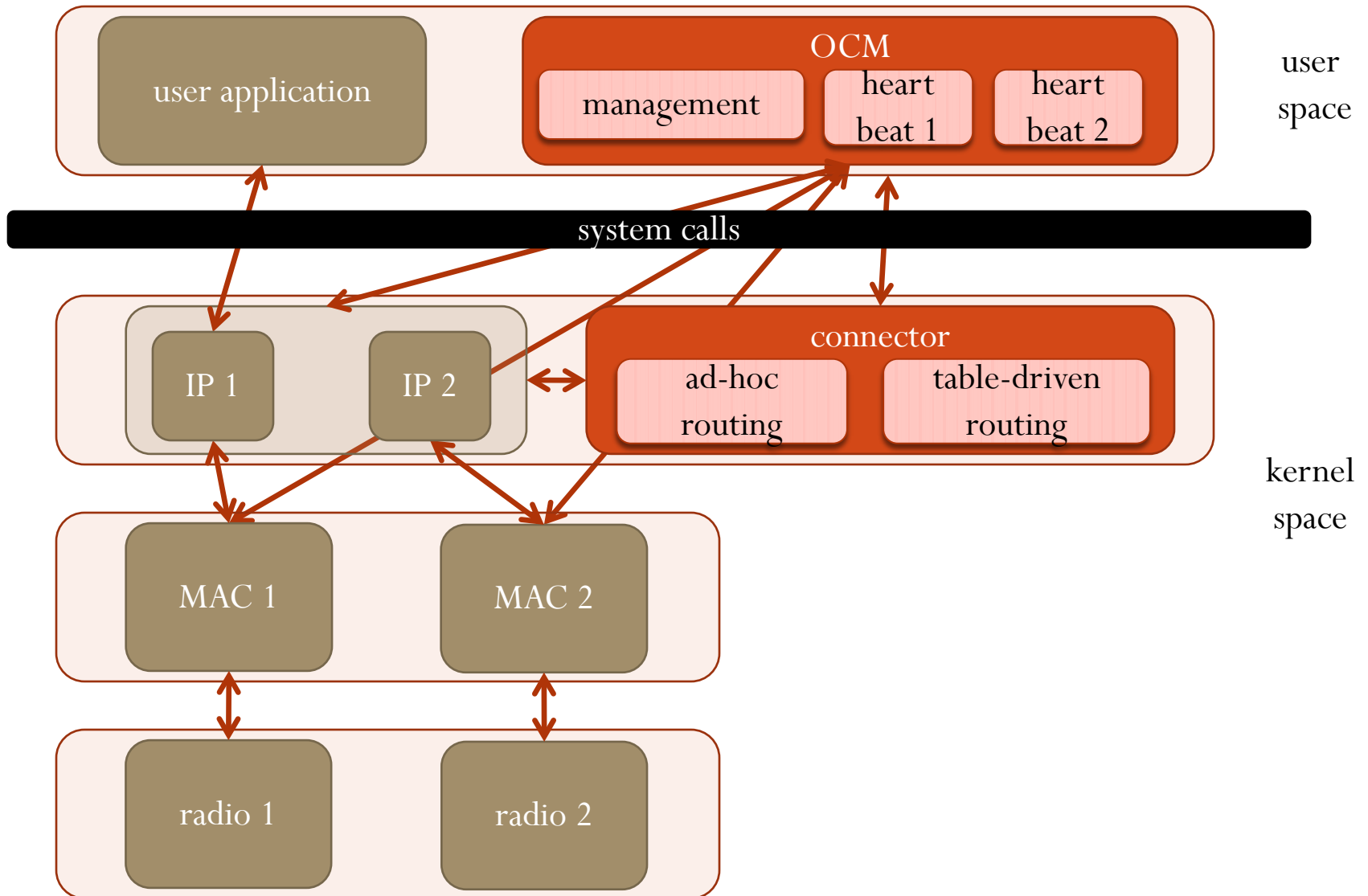
- Overlay-based approaches
  - overlay nodes are responsible for routing, other turn radios off
  - but, connectivity is lost
  - overlay nodes buffer messages intended for their passive neighbors
    - increased number of failed deliveries / latency / storage
- Variable transmission range
  - decreased range → decreased power
  - but, idle power is still a problem
- Built for custom combination of radios
  - e.g., WiFi and BT
  - do not generalize for others (and even more than two)

# Our middleware

- Generic
  - does not assume any specific radio technology
  - presented for two, but easily generalizes for any number of interfaces
- Transparent for an application
- Ensures connectivity
- Limits the impact on latency and capacity
- Requires minimal modification of OS kernel
  - customized routing protocol at the IP level



# High-level architecture



# High-level architecture cont.

## Overlay Construction and Maintenance (OCM)

- Runs in the user space
  - does not delay packet processing in the communication stack
- Management module
  - contains the main logic of the middleware
  - constructs and maintains power-aware overlay
  - switches radios on and off via system calls
- Heartbeat modules
  - maintain the set of neighbors at each node
  - one module per radio interface

# High-level architecture cont.

## Connector module

- Slightly modified routing infrastructure
  - exposes a standard routing API for the IP module
  - contains two routing protocols:
    - reactive ad-hoc routing
      - standard protocol, e.g., DSR, AODV, ...
    - proactive table-driven routing
      - routing table managed by OCM
- In linux, can be implemented by means of iptables configuration

# The goal of OCM

Denote two available interfaces by A (long) and B (short)

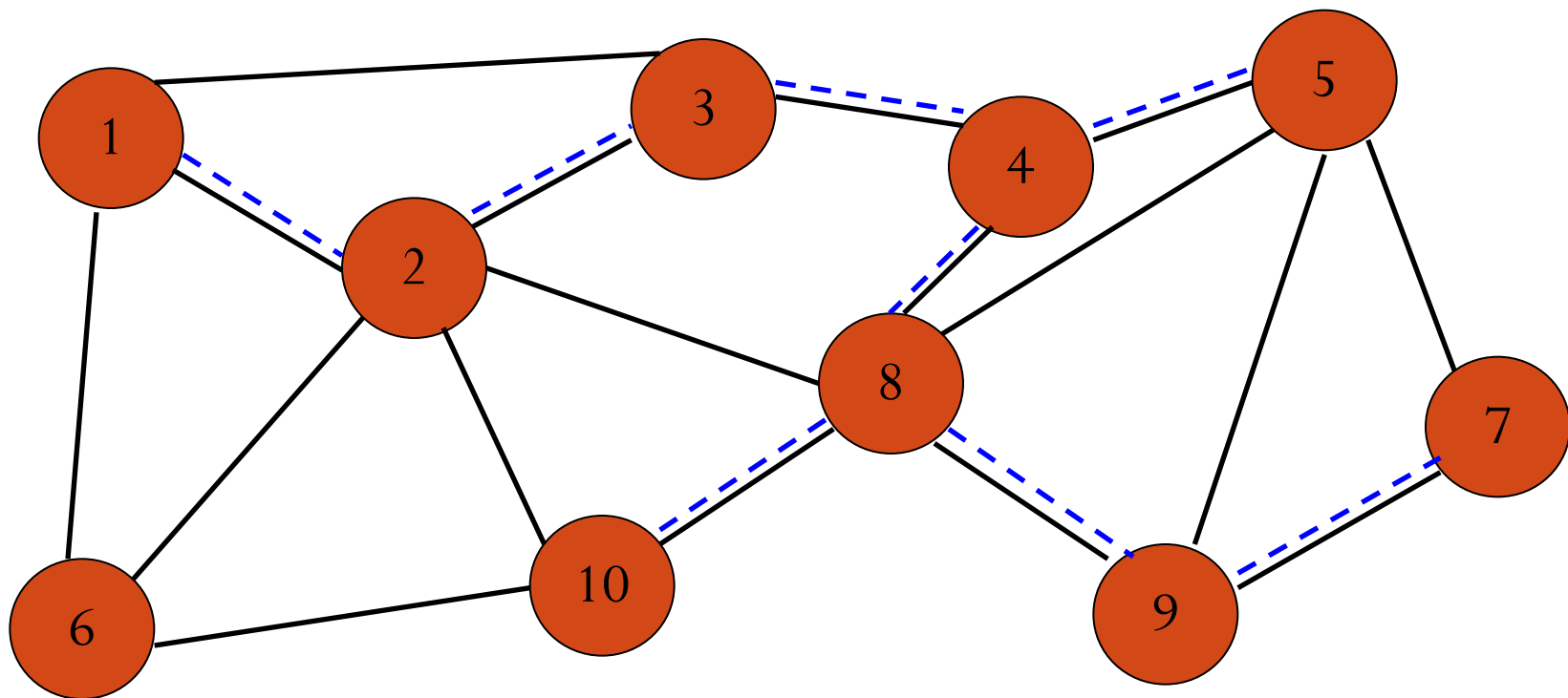
Select nodes into an overlay such that

- any non-overlay node is associated with some overlay node
- any non-overlay node turns its long-range radio A off

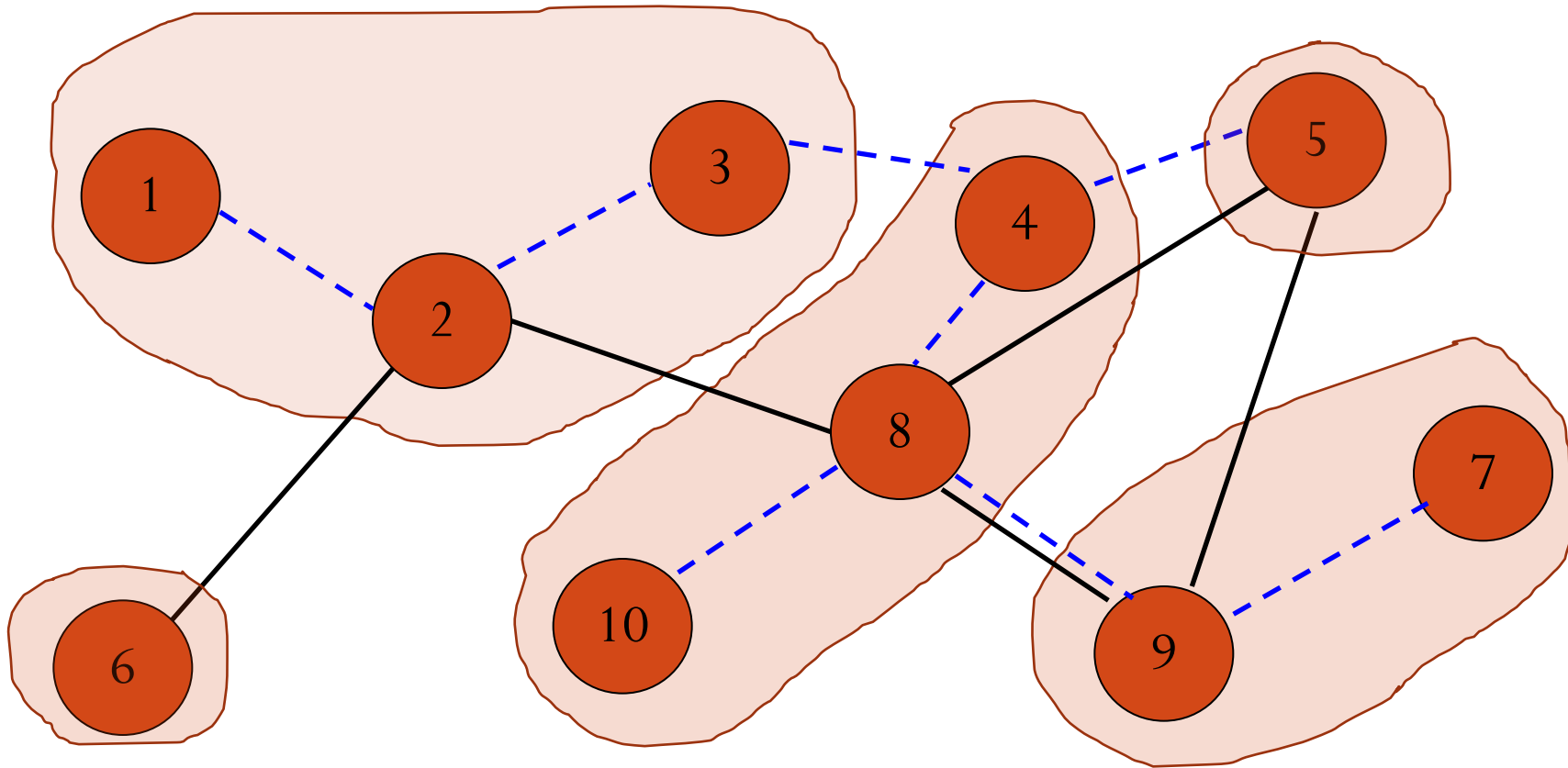
under the following restrictions:

- No device is further than  $k$  short hops from a device with an active interface A
  - limit the impact on network latency and capacity
- The sub-network formed by devices with active interface A is connected
  - the whole network is connected on at least one interface
- The devices with active interface A have high remaining energy
  - avoid overlay recomputation due to exhausted nodes

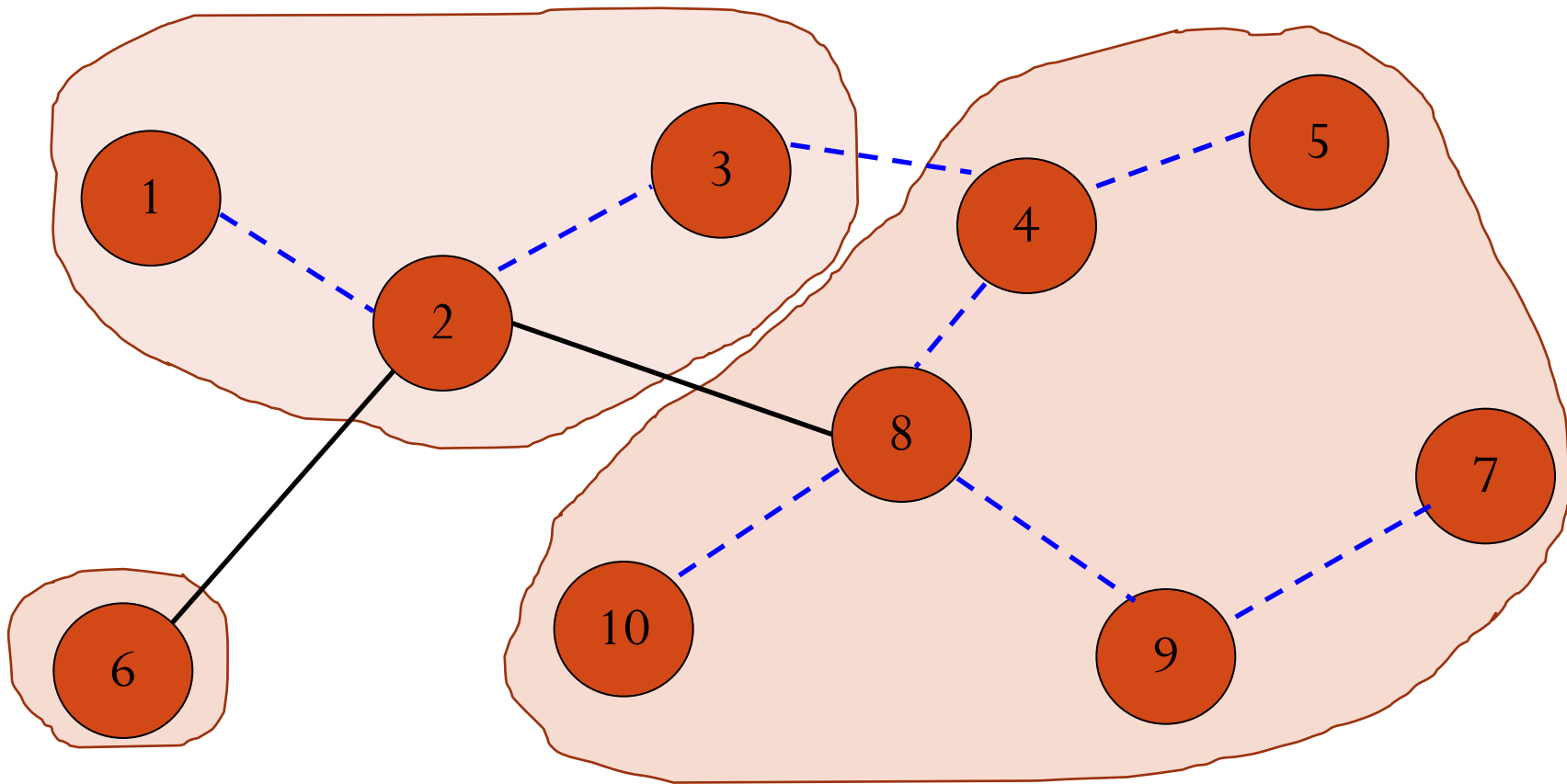
Example:



# Example: $k=1$

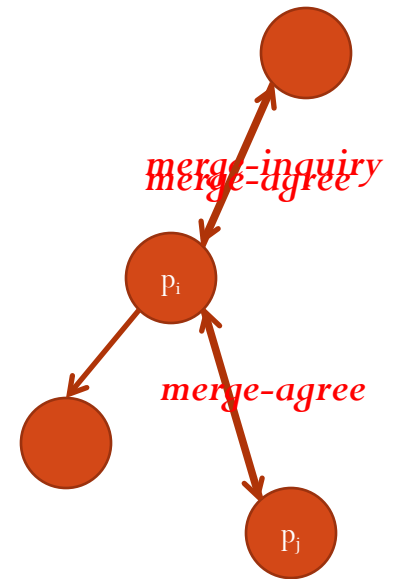


# Example: $k=2$



# The overlay construction protocol

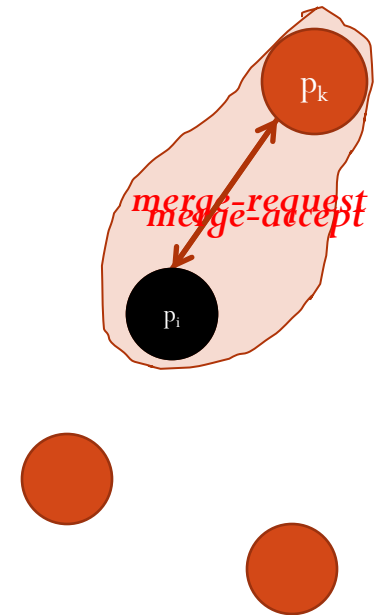
- Each node starts as a *cluster-head* of an empty cluster
- Each cluster-head  $p_i$  periodically publishes its cluster status to neighbors on interface A, and waits for replies
  - cluster status is composed of  $p_i$ 's energy level, current neighbors on interface A and B, and an intra-cluster routing table
- A neighbor  $p_j$  checks several sanity conditions before it agrees to absorb the cluster of  $p_i$





# The overlay construction protocol cont.

- From all neighbors responded positively,  $p_i$  chooses the one with the highest rank,  $p_k$ , and sends a request to merge
- If  $p_k$  did not send such request to some other node, it agrees
  - finally,  $p_i$  becomes a *peer* in the cluster of  $p_k$ 
    - turns its interface A off
    - notifies all peers in its cluster on the change of their cluster head



# The overlay maintenance protocol

- A cluster-head periodically broadcasts the intra-cluster routing table to its peers
  - “keep-alive” for the cluster head and short links
- A peer becomes a cluster-head again if:
  - does not receive any message from its cluster-head for too long
  - a path to the cluster-head is broken
    - notified by heartbeat module
  - cluster-head sends a retire message
    - when its energy level reduces significantly

# The integration with routing

- Given a packet to route, the connector module checks the destination:
  - if belongs to the cluster, use the intra-cluster routing table
  - if not, check the source
    - if a peer, use the intra-cluster routing table to route to the cluster-head
    - if a cluster-head, give the packet to the reactive ad-hoc routing
- A cluster-head runs an ad-hoc routing protocol, while serving as a proxy for its peers
  - responds for route requests sent for peers in its cluster
  - routes any packet intended for its peer through the intra-cluster routing table

# Performance evaluation

- Performance evaluated in SWANS
  - Java-based network simulator developed at Cornell U.
    - extended with energy model and multiple interfaces support
  - full implementation of the described architecture
- Simulation area of size  $500 \times 500 \text{m}^2$ , 100-1000 nodes, 1000 (real time) seconds
- Each data point produced based on 10 experiments
- Each node equipped with BT and WiFi radio
  - BT has 10m transmission range
  - WiFi has 100m transmission range

# Energy model

- Energy model of WiFi and BT cards based on measurements reported by several papers

Mode of operation	WiFi	BT
Transmitting	1346mW	81mW
Receiving	900mW	81mW
Idle	740mW	5.8mW
Sleeping	47mW	not used

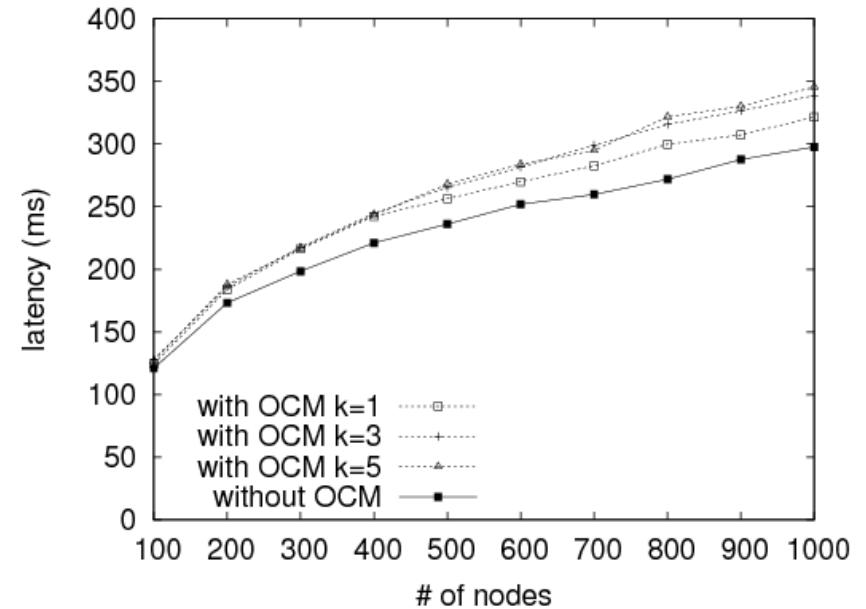
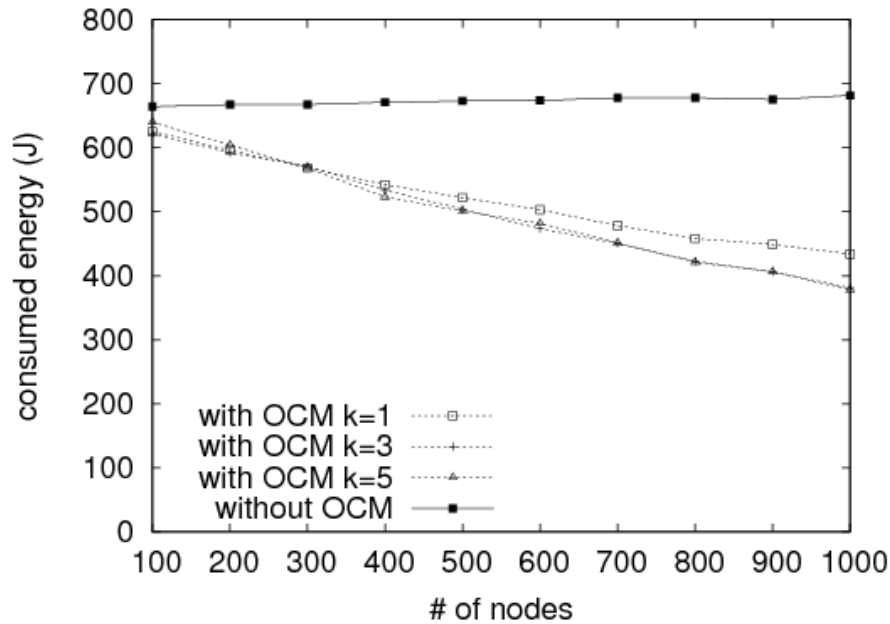
- In the ongoing work, we are validating the numbers and getting similar results



# Mobility models

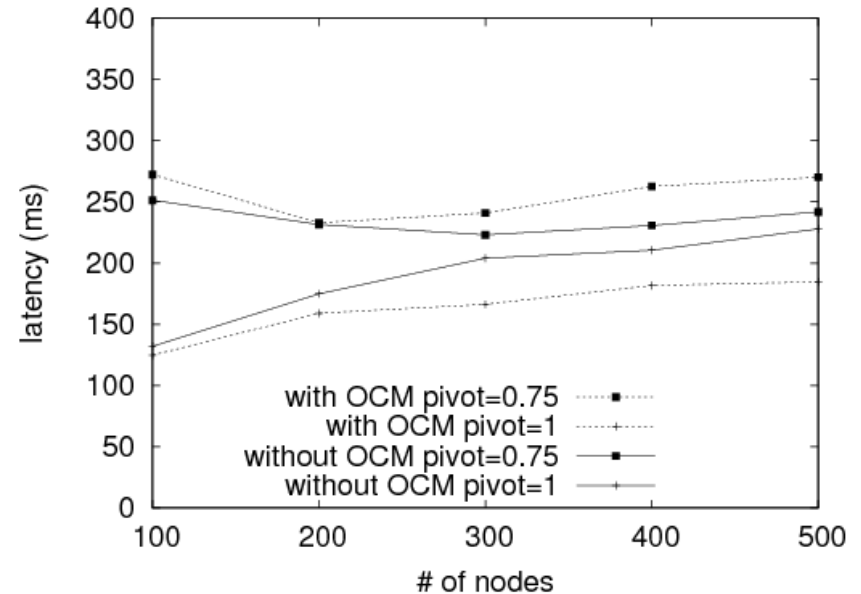
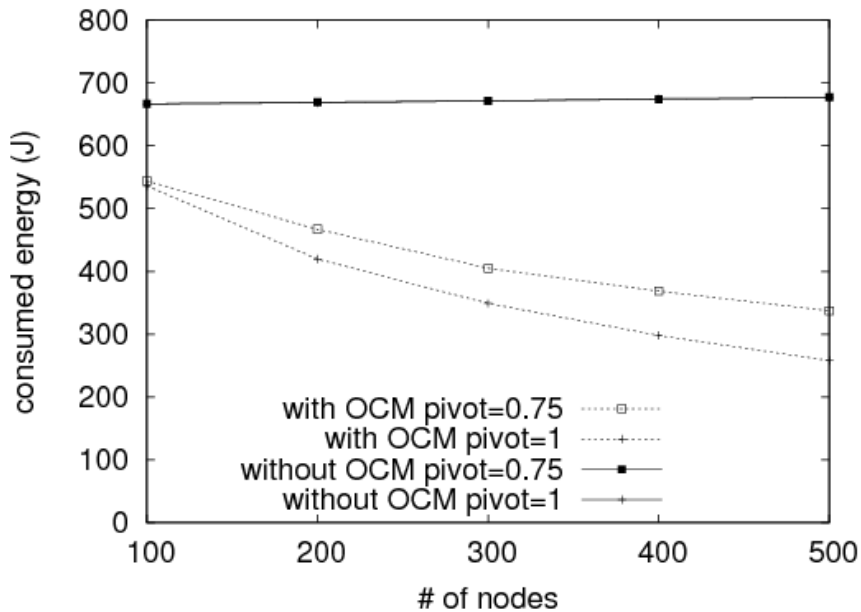
- Static
  - nodes remain at initially random positions
- Random way-point
  - loop through pausing for  $X$  seconds, then choosing a random location and moving there at a fixed speed
    - experimented with  $X=60$  and  $180$  seconds and walking speeds between  $1$  and  $2\text{m/s}$
- Two-phase model with hot-spots
  - captures the environment of a campus or a school
  - two sets of hot-spots designate preselected locations
    - one set for class rooms
    - one set for gathering places, e.g., cafeteria, library, labs, etc.

# The impact of $k$ on the performance



- Even with  $k=1$ , the power savings are significant
  - and increase linearly with the number of nodes
  - further increase in  $k$  does not help
    - due to limited connectivity at the BT level
- A moderate impact on latency

# Two-phase mobility with hot-spots



- More than 50% of energy saved
  - high efficiency is achieved due to concentration around hot-spots



# Summary

- OCM achieves significant energy savings
  - especially, when nodes are concentrated around hot-spots
  - latency and message loss rate are almost identical to a standard ad-hoc routing algorithm
- Adapts well to the density of the network
  - exhibits “add more to improve service” behavior
- Generalizes easily for an arbitrary number of interfaces
- Next: evaluate OCM in a real setting of mobile phones

Thank you!

Questions?