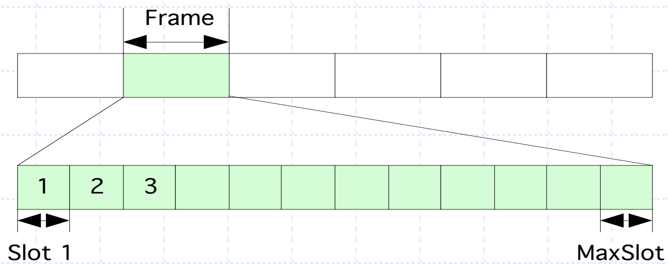
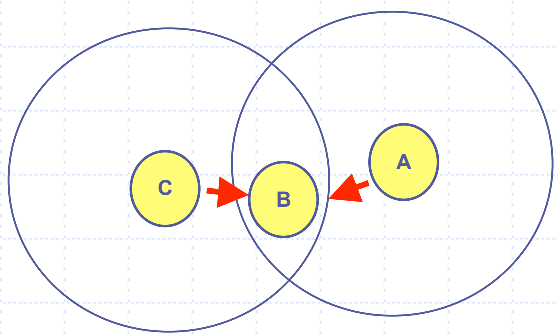
DRAND: Distributed Randomized TDMA Scheduling for Wireless Ad-Hoc Networks

Injong Rhee
(with Ajit Warrier, Jeongki Min, Lisong Xu)
Department of Computer Science
North Carolina State University

Static Channel Assignment Problem (for stationary networks)

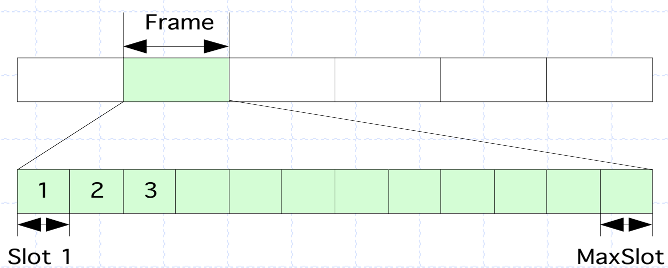
- Finding a time slot for each node such that any two nodes within an interference range do not have the same transmission time slot.
- G=(V,E), V: set of nodes, E: set of edges
  - Edge e =(u,v) is exists iff ***u* and v can hear each other.**

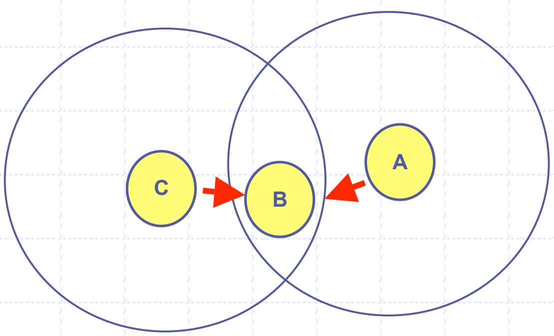




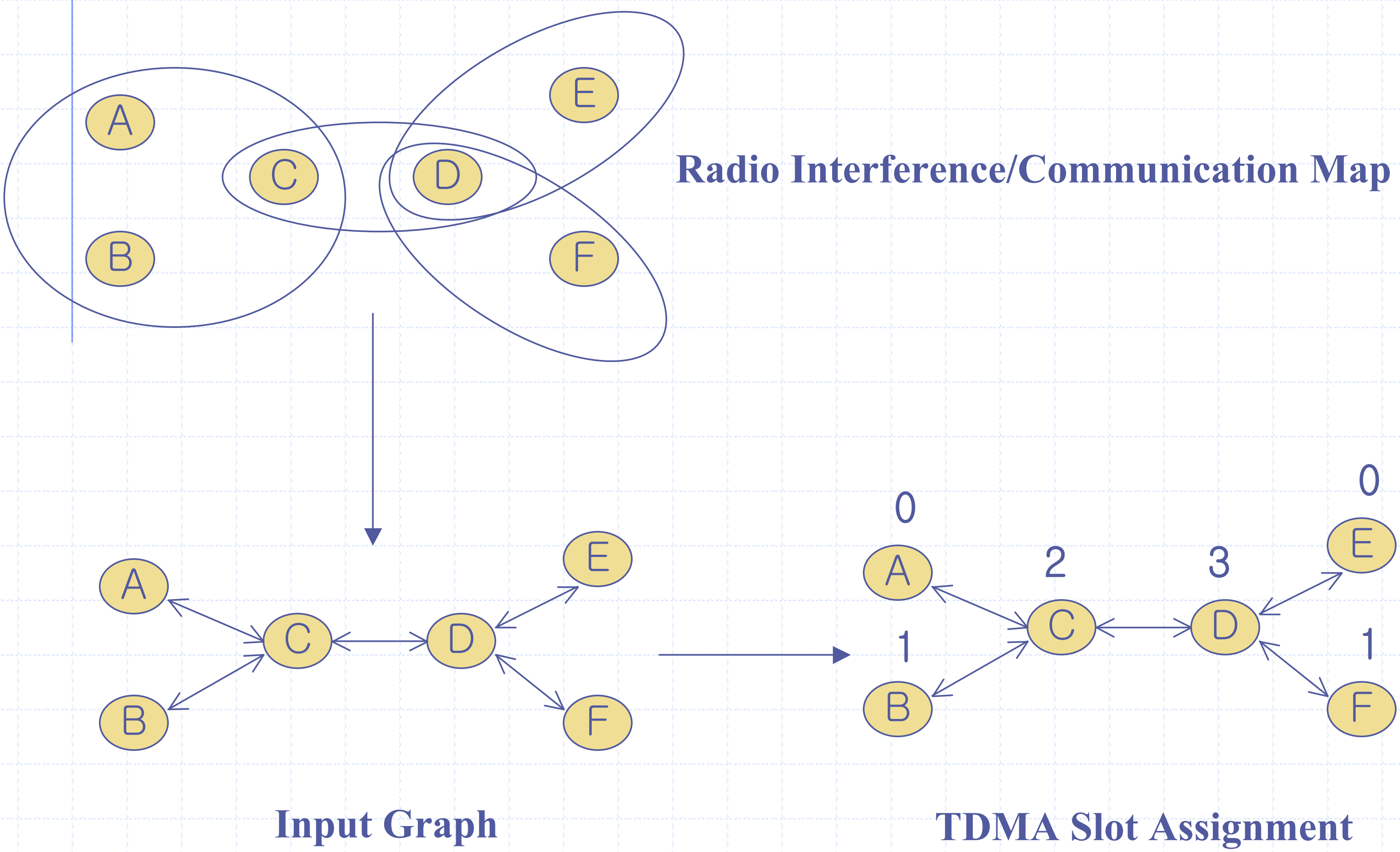
Announcements

- Finding a time slot for each node such that any two nodes within an interference range do not have the same transmission time slot.
- G=(V,E), V: set of nodes, E: set of edges
  - Edge e =(u,v) is exists iff ***u* and v can hear each other.**





TDMA Scheduling - Example (Broadcast: 2 hop interference)



Performance goals for TDMA Scheduling

- Efficient: Use as few slots as possible. More slots would imply less spatial reuse and longer delay.
- Distributed: The algorithm must not require global information or global coordination of any kind.
- Simple: Low Time/Message Complexity. The algorithm must be simple to implement.

RAND by Ramanathan [Infocom97]

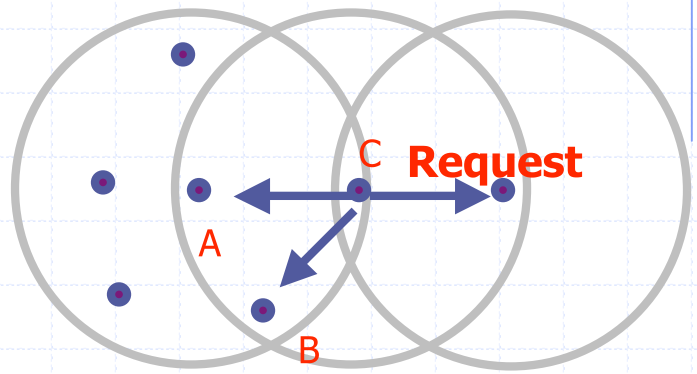
- TDMA scheduling is a coloring problem.
  - Hence Optimal TDMA scheduling is **NP-Hard**.
- RAND: So a heuristic, but centralized algorithm:
  - Total-order all the nodes in a random order.
  - Assign to each node in that order the minimum color that has not been been assigned to its two-hop neighbors in the graph.
  - Gives "pretty" good efficiency: at most d+1 colors, but mostly much fewer colors (d is the number of conflicting neighbors).

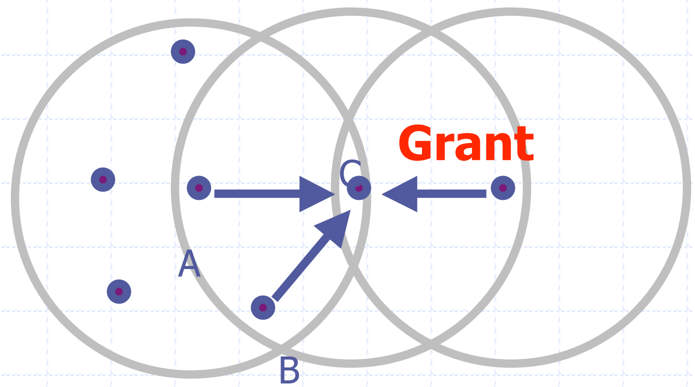
DRAND: Distributed RAND

- Performance
  - as efficient as RAND
    - The first distributed version of RAND
  - Simple, Running time/msg complexity O(d) – d is the max number of contenders (two-hop nodes)
- Key assumption
  - Each node knows its neighbors.
  - Packet losses are possible
  - No time synchronization is required.
- Key idea
  - When a node is selecting a color, it ensures that no other neighboring nodes are selecting.

DRAND: How it works

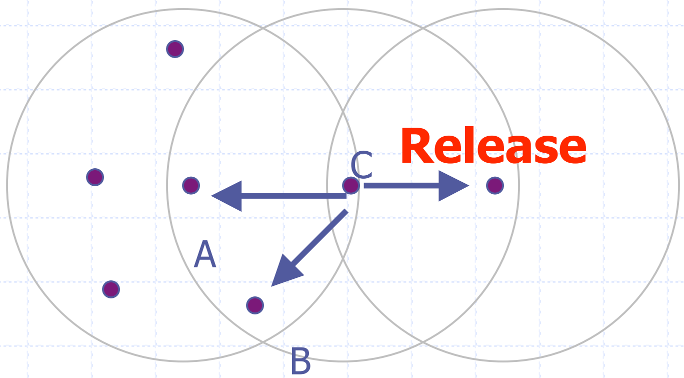
- Algorithm runs by rounds
- With some probability, node A sends *request* message if A does not have time slot.
- Neighbor B sends *grant*(containing its one-hop neighbors' slot info) if it is not aware of any of B's neighbors that has sent a request.





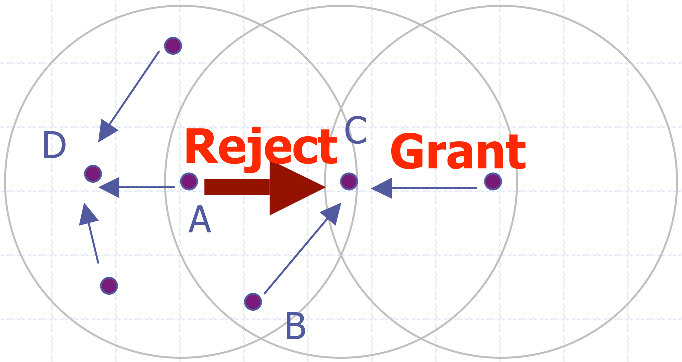
DRAND: How it works (cont.)

- If and when A receives *grant* messages from its entire one-hop neighbors,
  - it chooses the minimum of the time slots that have not been taken by its two-hop neighbors, and then broadcast a release message.

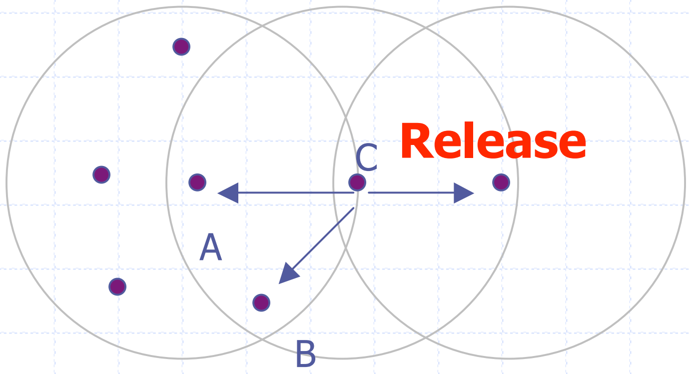


DRAND: How it works (Cont.)

- When a node has granted to another node, it sends back a reject.



- If a node receives reject or timeout, then it sends release (but with failure indication).



DRAND – Complexity Results

- Running time is O(d) – d is the maximum number of nodes within two hops for the entire graph.
- Message complexity is O(d).
- Achieves the same slot efficiency as RAND

DRAND Experimental Evaluation Methods

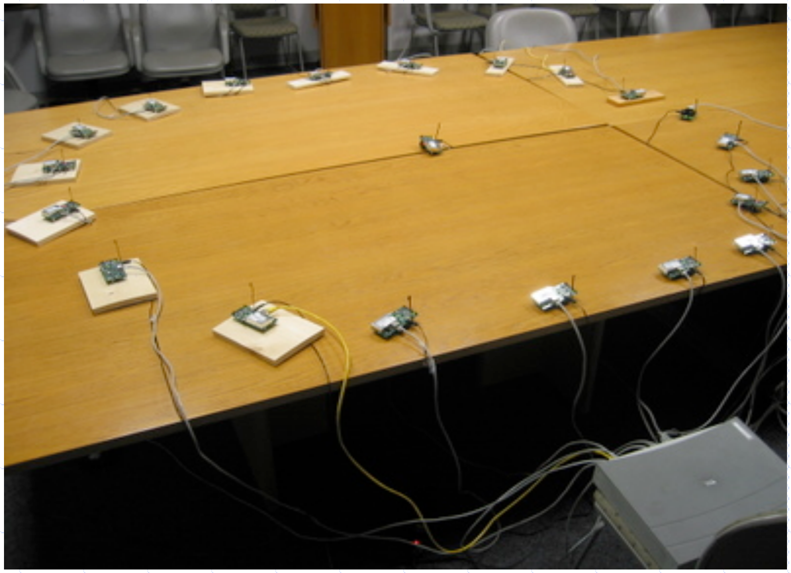
- Verification of analysis
- DRAND performance overhead on a small scale mote test bed
- DRAND performance comparison with existing TDMA assignment schemes

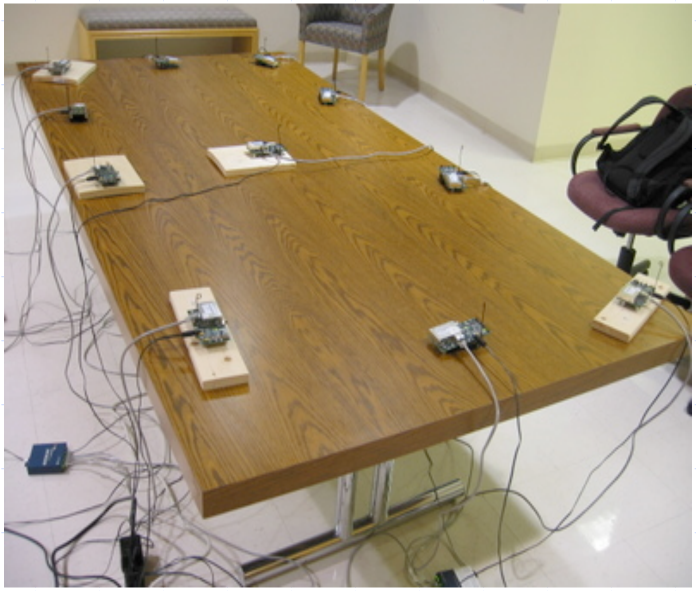
Experimental Setup – Single/Multiple Hop

Single-Hop Experiments:

- Mica2 motes equidistant from one node in the middle.
- All nodes within one-hop transmission range.
- Tests repeated 10 times and average/standard deviation errors reported.

NS simulation (random Poisson point model)

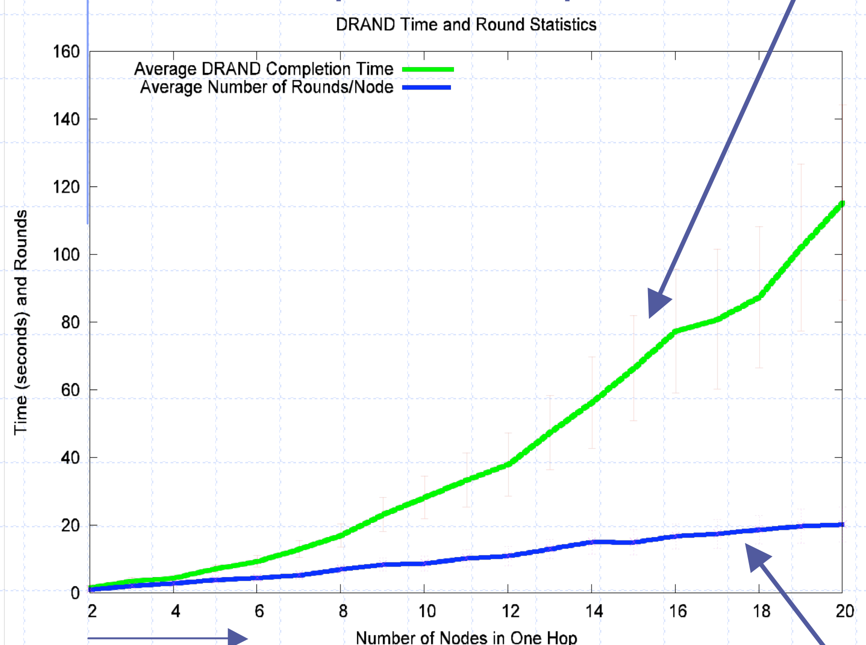




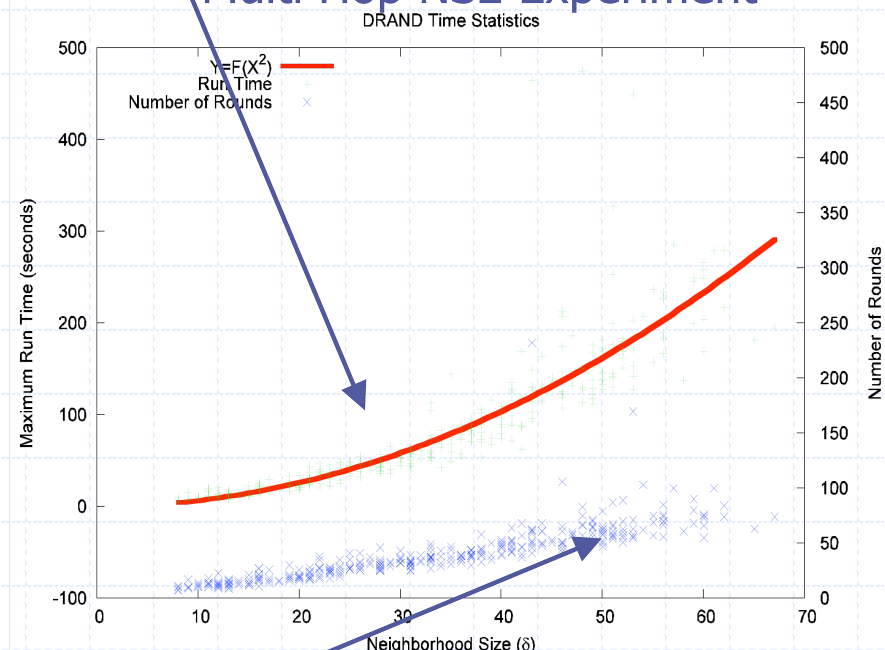
DRAND – Time Complexity

Running Time

One Hop Mica2 Experiment



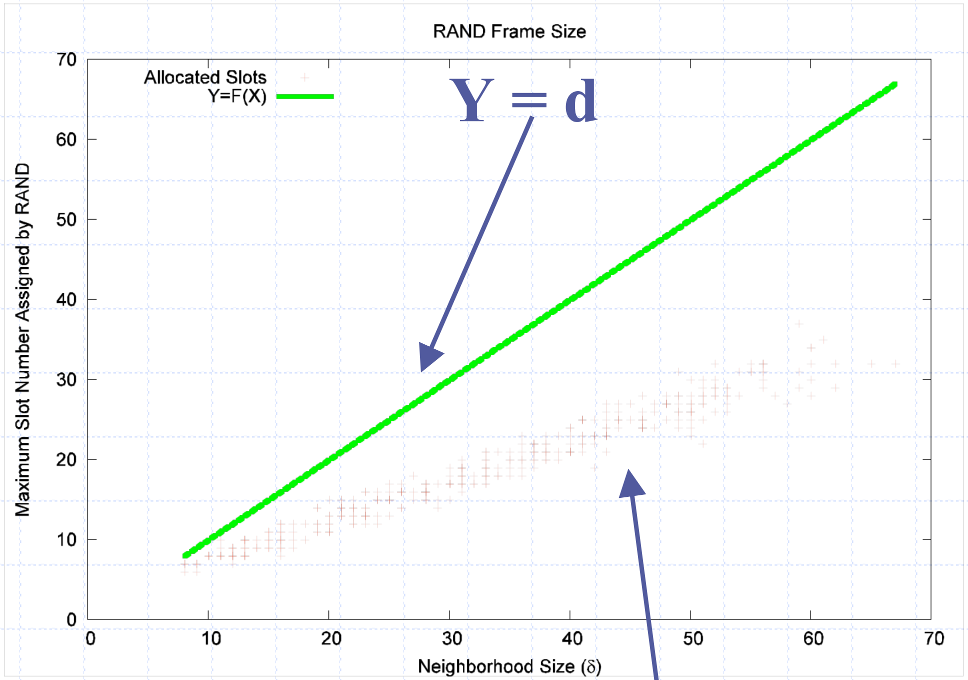
Multi Hop NS2 Experiment



Num of neighbors

Rounds

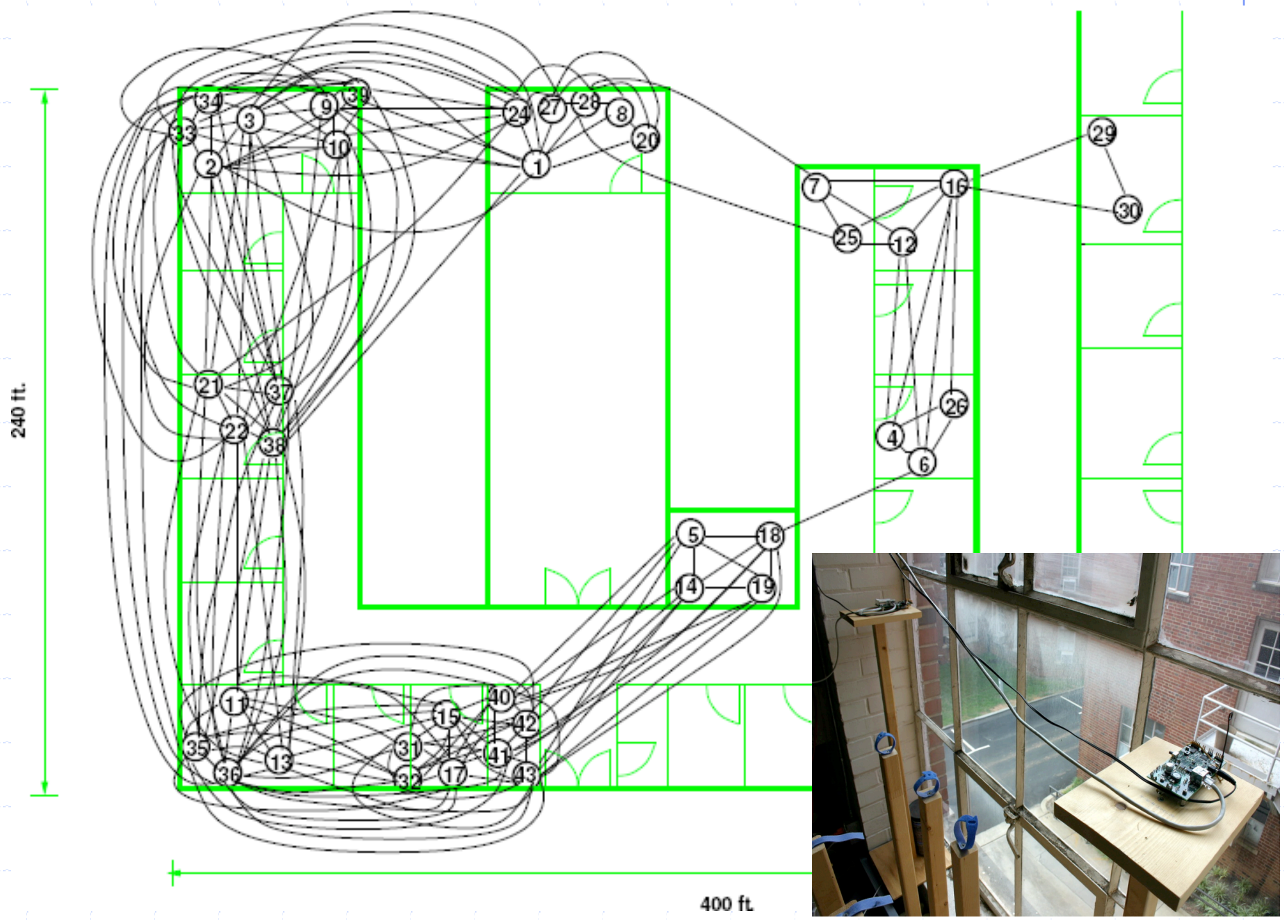
DRAND – Maximum Number of Slots



Maximum number of slots in the multi hop NS simulation

Experimental Setup - Testbed

- **40 Mica2 sensor motes in Withers Lab.**
- **Wall-powered and connected to the Internet via Ethernet ports.**
- **Programs uploaded via the Internet, all mote interaction via wireless.**
- **Links vary in quality, some have loss rates up to 30-40%.**
- **Asymmetric links also present.**



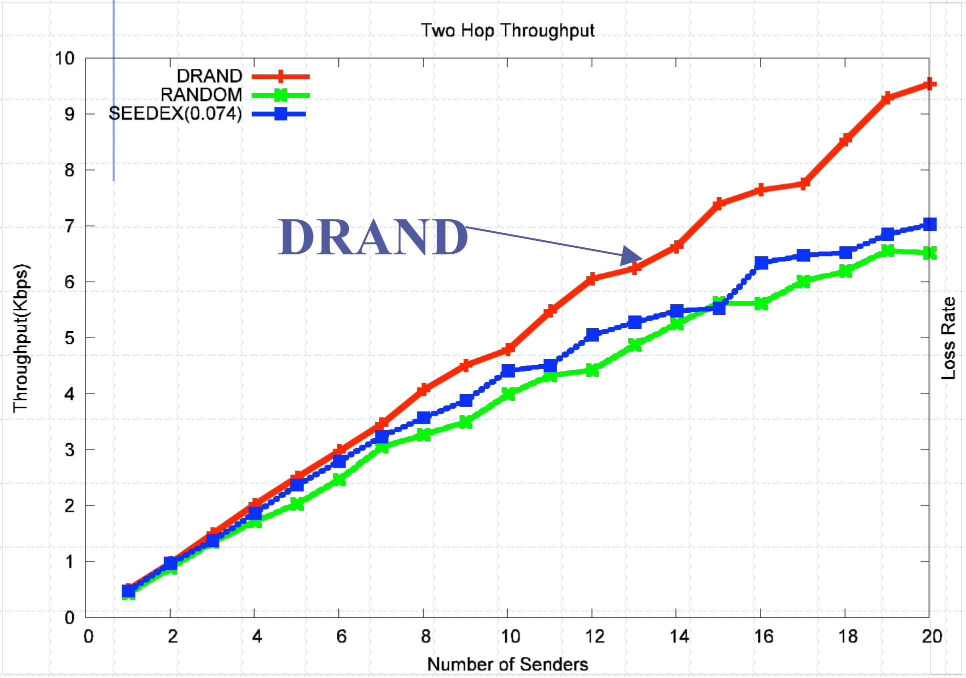
DRAND – Time and Energy for each phase

Operation	Average Running Time	Average Energy Cost
Neighbor Discovery	30s	0.732J
DRAND Assignment	194.38s	4.88J
Slot Dissemination	60s	1.33J

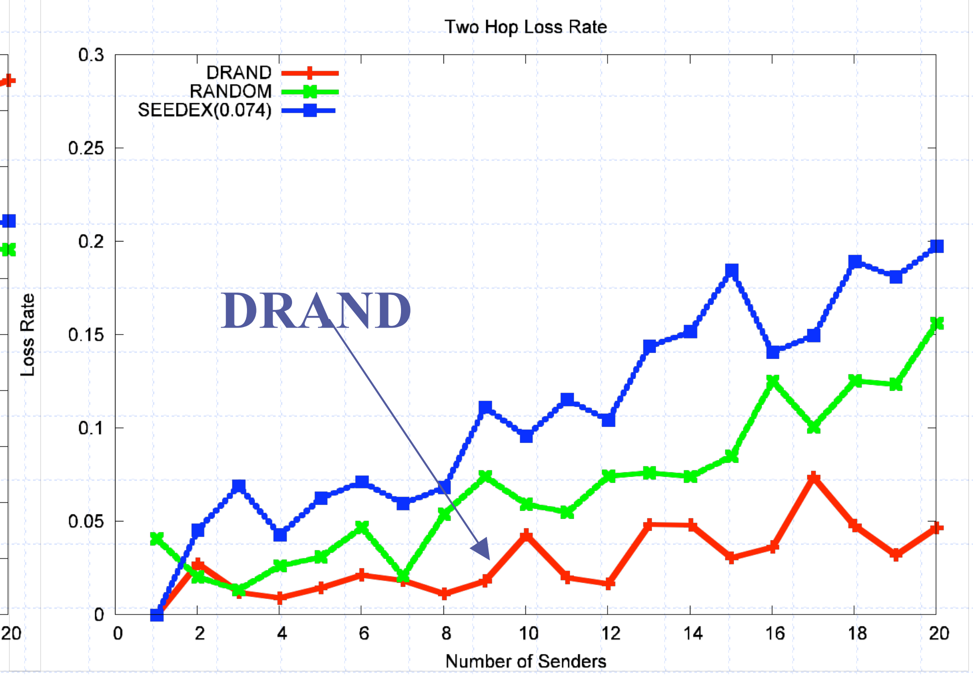
Total Energy (6.942J) is 0.02% of the total battery capacity of a node with 2500mAh and 3V.

Comparison with existing TDMA schemes: Throughput/Loss Rate

DRAND **RANDOM** **SEEDEX**



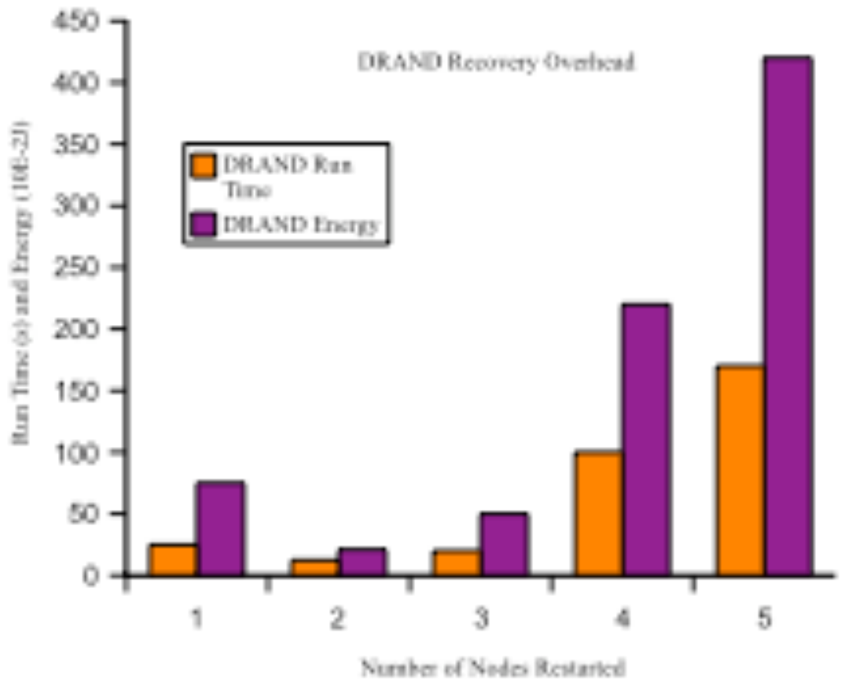
Throughput



Loss Rate

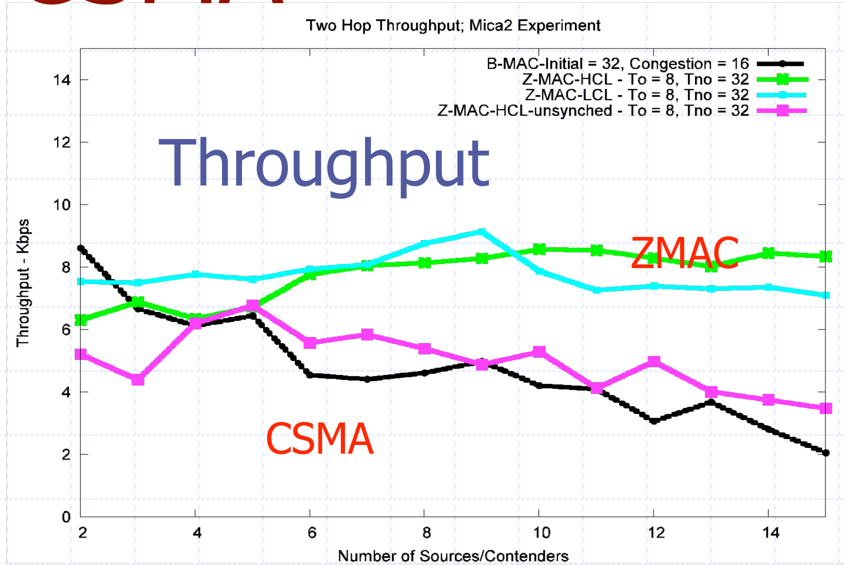
DRAND - Adapting to Changes

- As new nodes join or topology changes, it requires only local adjustment.
- We measure the overhead as nodes restart after crash.

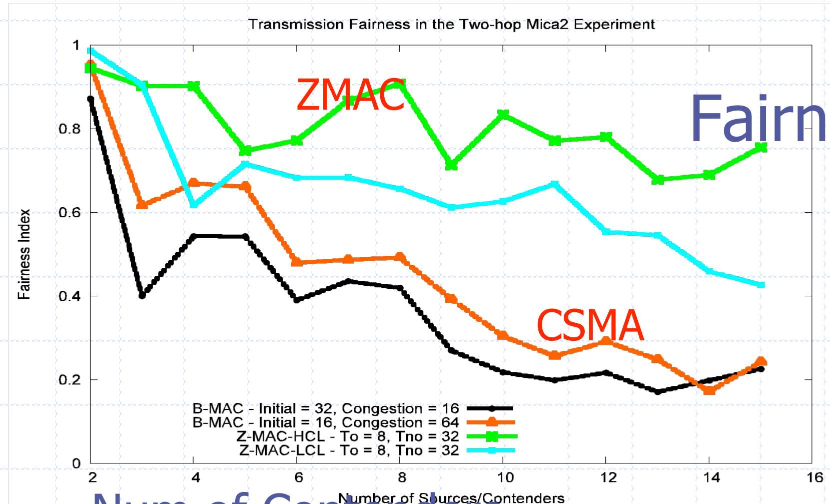


Comparison with CSMA

- ZMAC [Sensys05].
- A hybrid of CSMA and TDMA
  - High throughput
  - Highly fair (reduce a lot of location and topology dependency)
- Implemented over IEEE802.11.



Num of Contenders



Num of Contenders

Conclusion

- DRAND is a distributed implementation of RAND
  - Efficiency, Running time and Msg Complexity : O(d)
- Can also be applicable to other problems such as
  - ID assignments
  - Frequency channel assignment.

Experimental Topologies for Verification of Analysis

- Report DRAND **running time, message complexity, maximum number of slots** assigned for each run.
- **One Hop Mica2 Experiments**
  - 20 Mica2 motes within radio range of each other.
- **Multi-Hop NS2 Simulations**
  - 50-250 nodes placed randomly on a 300x300 grid, creating node densities between 5 and 60.
  - Radio range 40m
  - Bandwidth 2Mbps
  - 802.11 MAC

Support Slides

Performance Comparison with Existing Schemes

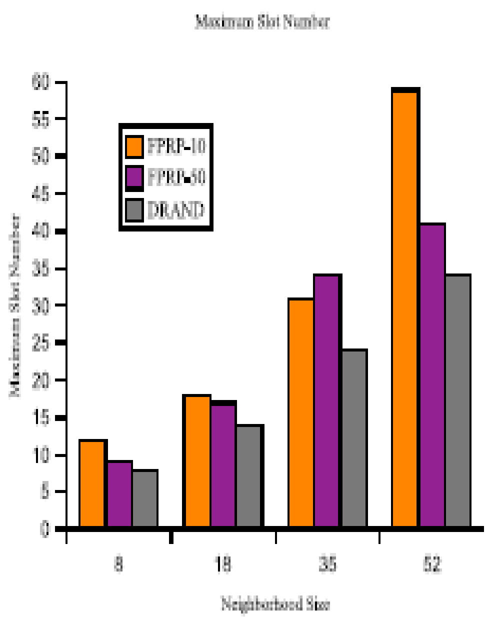
- Compare:
  - **Algorithm Complexity (on NS2 topology)**
    - Number of slots assigned
    - Run time
    - Number of messages transmitted
  - **Transmission efficiency (on two-hop topology)**
    - How well other schemes utilize the TDMA slots assigned to them

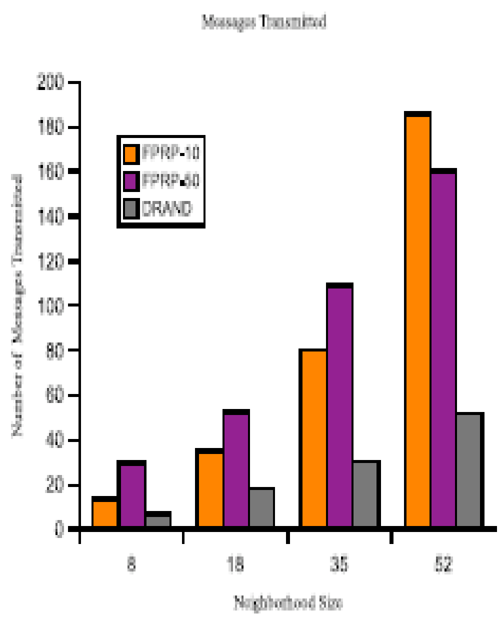
Number of Slots Assigned

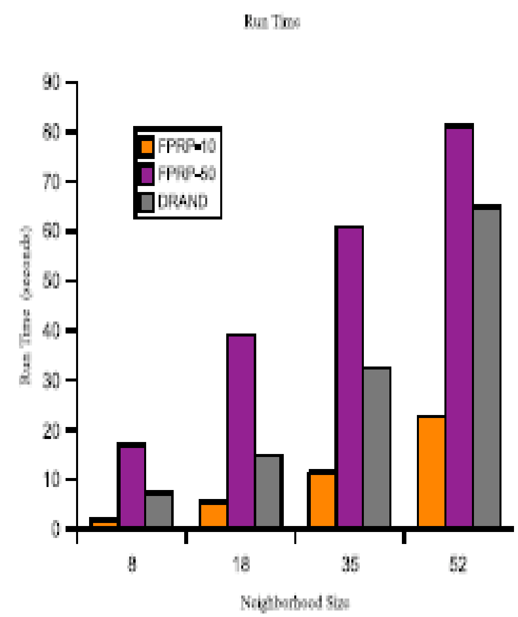
- FPRP (Five Phase Reservation Protocol)
  - Series of reservation/transmission phases
  - For each time slot of transmission phase, run five phases for a number of times (cycles) to pick a winner
  - FPRP-x indicates that we run FPRP for x cycles to determine the ownership of each slot
  - More cycles => more nodes get a slot, but correspondingly increases running time

Algorithm Complexity – FPRP and DRAND

FPRP-10 **FPRP-50** **DRAND**





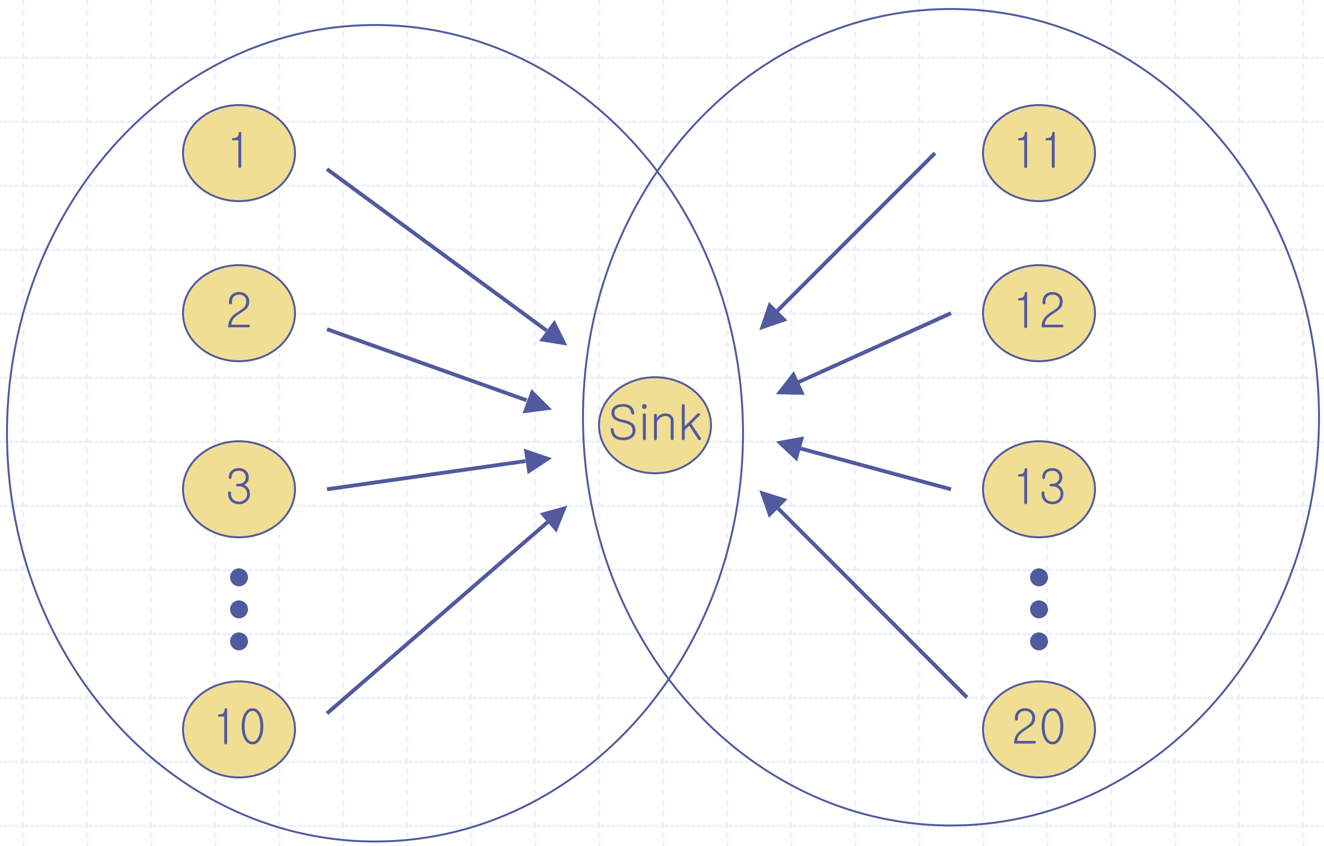


Maximum Slot Number Message Transmissions Run Time

Transmission Efficiency

- **SEEDEX**:
  - Nodes send in a slot with probability P.
  - Random number generator seeds of every node in two hop neighborhood known, through message passing
  - Hence each node knows the number of nodes eligible to send in a slot = C
  - Send in that slot with probability 1/C
- **Randomized Slotting**:
  - At the beginning of the frame, select slot randomly
  - Transmit

Two Hop Mica2 Topology for Transmission Efficiency Experoments



Interference in Wireless Networks

- **Primary Interference**
  - A station cannot transmit/receive at the same time
- **Secondary Interference**
  - Two stations within radio range cannot transmit at the same time (**single-hop interference**)
  - Two stations within radio range of a common node will cause collision if they transmit at the same time (**hidden terminal problem**)

Scheduling in Wireless Networks

- **Broadcast Scheduling**
  - The stations themselves are scheduled
  - The transmission of a station must be received collision-free by all its one-hop neighbors
- **Link Scheduling**
  - The links between stations are scheduled
  - The transmission of a station must be received collision-free by one particular neighbor.