## Mobile Sinks and 2BD 1

Deal with **isolated region**s (sparse WSNs)

Relax constraints on network connectivity -> fewer nodes -> reduced costs

Mitigate the funneling effect (interference and collisions)

**Challenges and opportunities**

Detection of Mobile Sinks (discovery problem)

Mobility-aware power management

Reliable data transfer

Mobility control and optimization

**Sensors/Sinks interaction**

Discovery & data transfer phases

Mobility can be deterministic and random

**Discovery process**

**Scheduled:** specific instant to be in contact, simple to implement but tight synchronization **On demand:** sensors wake up by the MS, use multi radios like RFID to trigger the activation **Asynchronous:** duty cycles, periodic listening, parameters have to be properly defined to ensures MS be discovered. Distance and speed impact MLP

Contact time (discovery time and residual time)

**Mathematically determine ON and OFF`**

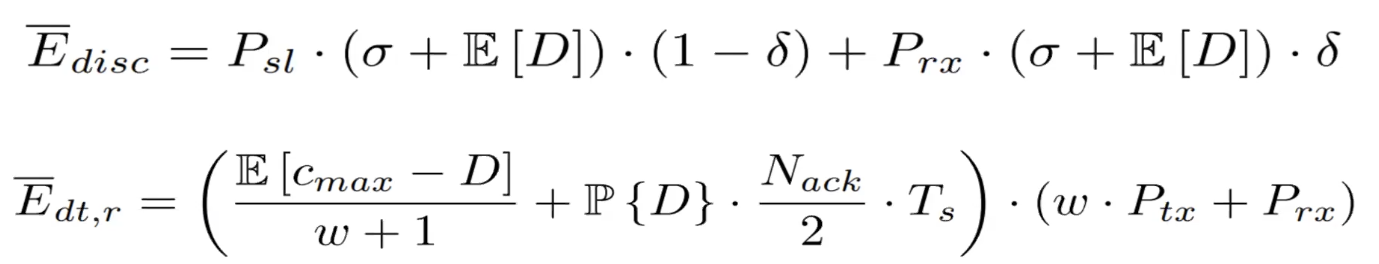
**Markov Chains:**  
 Stochastic model; a sequence of events; next event only depends on previous state; DTMC discrete time

**Stationary distributions:**

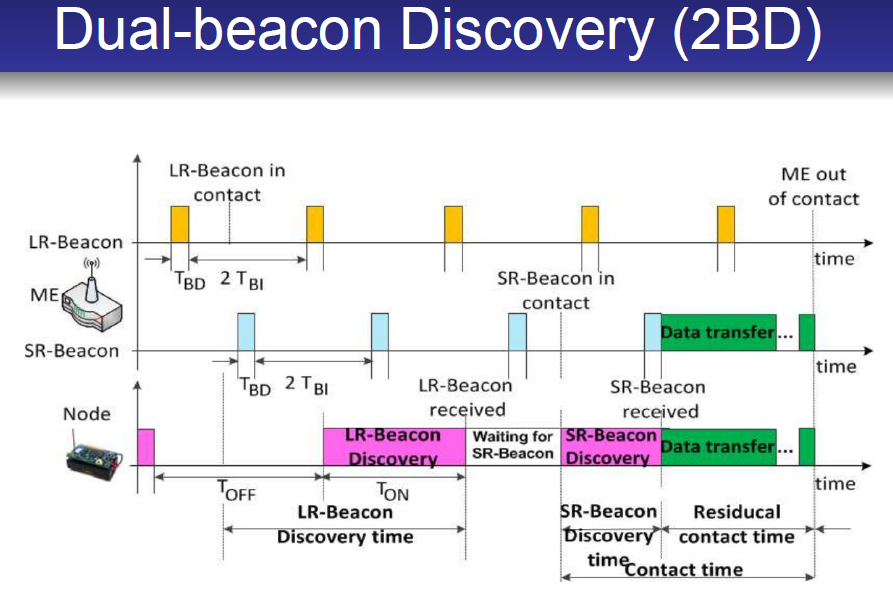
Probability that remains unchanged absorbing state

Data transfer analysis (single window, loss prob changes)

Energy model



**2BD results:**



loses performance some cases (duty cycle low < period < high less waiting time)

larger discovery area better

## Pros and Cons of PL and 2BD:

2BD: complicated to model

Longer waiting time better

PL: Simple algorithm

Waiting time is short

## **SISSA algorithm**

One Beacon, one duty cycle per node Using beacon to sync 36

Each node has a node id, TDMA scheme

One Beacon -> sync with each other -> Swarm agents broadcast (Residual energy)

**Swarm** and **communication** phase:

1. Cannot converge until every sensor receives a swarm agent from others

Each sensor node will terminate the swarm phase at the same time

They know **when the swarm phase end**s

1. The sensor radio remains active **only** during the instants of swarm transmission/reception.

Worst-case convergence is bound;

Number of messages is constant

Min channel time, max energy consumption

Energy consumption overhead higher duty cycle <-> lower energy spent (worst case)

**Pros and Cons of SISA:**

Energy **low** throughput **high**

## Network Layer and Ad Hoc 2

**Routing in Ad Hoc networks**

**ID-centric**: Arbitrary node -> arbitrary node

Routing & Forwarding:

R: construct tables on how can be reached

F: which physical interface to forward to

**Challenges**: Optimize metric/Mobility changes relations/scalability

Standard routing approach not applicable: (need to know all nodes)

Large overhead, slow in reacting to changes

Simple solution: Flooding (overhead is prohibitive/energy consumption/waste bandwidth not scalable)

Ad Hoc Classification:

Proactive: always maintain fresh information about route table-driven

Reactive: route is only determined when needed on-demand

Hybrid

Classification 2: arbitrary identifier / geolocation / some structure

DSDV: 8

DSR: 9 Dynamic source routing & modifications -> stale route caches -> cache manage

AODV: 12 Ad Hoc On demand Distance Vector page energy too much

# Routing Protocols Geo routing

Location based (Geographic) protocols 18

Geographic routing: infer information from physical placement of nodes

Position-based (Most forward within range R / Nearest node with any progress(reduce energy) / Directional routing

**Pros and Cons Geo routing:**

**P:** Do not need to keep routing table simplicity

Scalability, memory don not increase over time

**C:** need to store minimal geo information, right trade off take energy need a centralized optimal solution or approximate solution (not best)

Location update frequently need to keep up overhead

Doesn’t work indoor

## Motivation NP PRADA:

Traditional Ad hoc routing algorithms are not scalable (too much energy, flooding) 39

Two big families: Hierarchical Routing || Geographical Routing

Topology control: cost of **information** & cost of **communication**

P vs NP: Energy model 33 NP Complete don’t know if the problem is solvable in POLY time

PRADA algorithm: select range from feedback

# Data centric and Hierarchical protocols LEACH

Content of data content collect process store in network

Data aggregation accuracy completeness latency Message overhead

Hierarchical protocols: to address scalability and energy consumption

Sensors nodes from clusters: cluster head form another layer divide and conquer

**LEACH**:48 random select as heads, minimize energy dissipation

Setup: random number (new head broadcast / each node access through min head (TDMA) Steady: send to aggregate and wait for another head selecting

Tradeoffs: number of clusters: few->much energy many->many nodes to sink

**Pros and Cons:**

Simple algorithm

Number of clusters, how long steady last + topology may not have advantage

dynamic clustering brings extra overhead diminish the gain in energy consumption

# Transport Layer PSFQ 3

dependable data transport transmission errors

handles reliable data transport / flow control / congestion control / network abstraction

TCP/UDP cannot be used

**Reliable data retransmit:**

Low bit error -> end-to-end else local retransmit

At most cases, handle at link layer

**HHR** || **HHRA**: 14

Repeat a few times || ACK

Delivery blocks of packets: lot time to occupy PSFQ 17

Distribute block of packets from one sender to multiple receivers (sink to sensors)

Source pumps into net using broadcast -> inter-packet gap

If not in sequence -> previous local recovery (do not forward first by fetch)

Scarifies delay to ratio 收到所有再向下一级广播

**Congestion**

Sensors same event -> report || own data

Detection: locally -> overly simplistic -> growth rate

Interaction with MAC -> CSMA high utilization TDMA-> difficult to detect

Handle: 1.Drop 2.Control sending rate 3.control how many sending 4.aggregation

Control how many are sending: N nodes k/N prob Gur Game

Event-to-sink Reliability: event-driven -> number of pacs in interval I || R required number for reliable detect r>R -> detect || reporting frequency f || transport problem configure f to get R

Sink: measure ri -> update f based on n and f n = ri / R

Broadcast to all sensors.

ESRT: monitor local buffer level + congestion notification -> avoiding congestion

Performance: considerable impact on the service no one size fits all

ESRT Cons: no priority || don’t fit big size network