HIERARCHICAL ROUTING

In a hierarchical architecture, higher energy nodes can be used to process and send the information while low energy nodes can be used to perform the sensing of the target.

Hierarchical routing is mainly two-layer routing where one layer is used to select cluster heads and the other layer is used for routing.

Examples include:

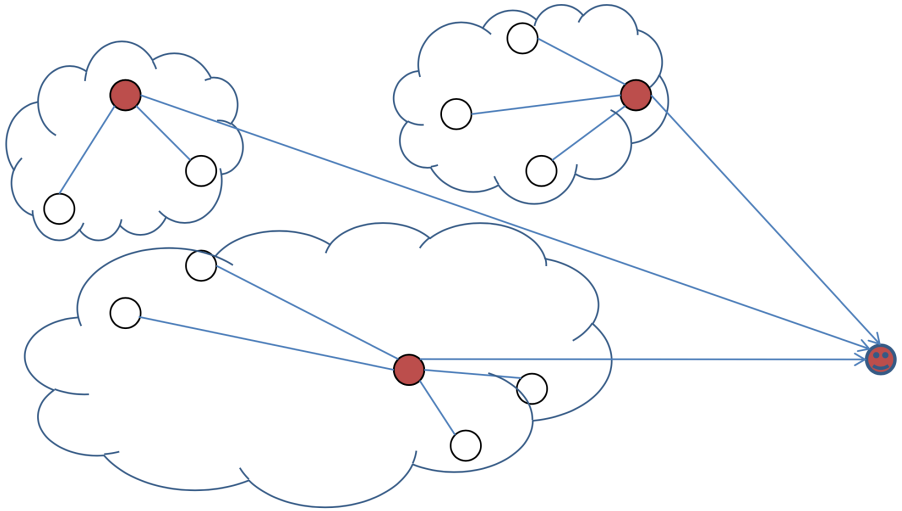
- LEACH (Low Energy Adaptive Clustering Hierarchy)
- PEGASIS (Power-Efficient Gathering in Sensor Information Systems)
- TEEN(APTEEN) (Threshold-Sensitive Energy Efficient Protocols)

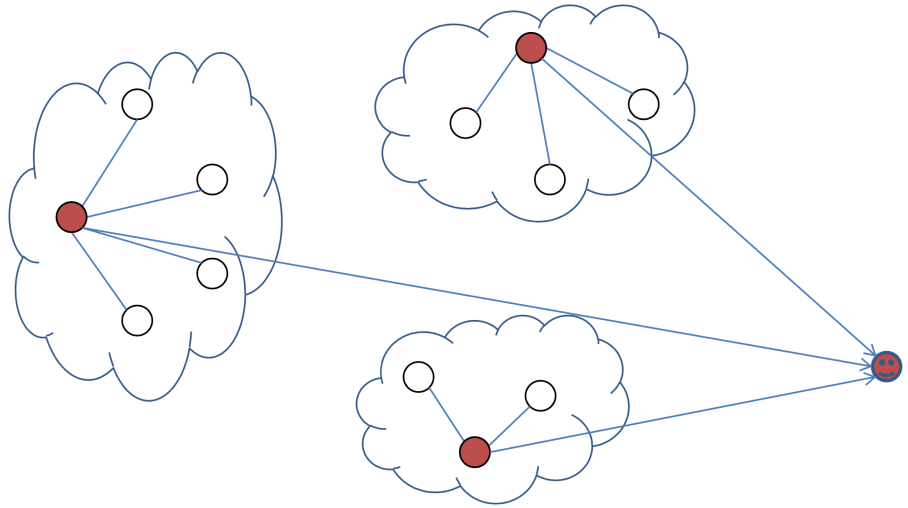
LEACH ROUTING

LEACH (LOW-ENERGY ADAPTIVE CLUSTERING HIERARCHY

LEACH Protocol Architecture LEACH in brief

- Randomized rotation of cluster heads among the sensors
- All non-cluster head nodes transmit data to their cluster head
- CH receives this data and performs signal processing functions on the data and transmits data to the BS





LEACH PROTOCOL

LEACH Protocol

- Cluster Head Selection
- Cluster Formation
- Steady State Phase
- LEACH-C: BS Cluster Formation

CLUSTER HEAD SELECTION ALGORITHM

- $P_i(t)$ is the probability with which node i elects itself to be Cluster Head at the beginning of the round r+1 (which starts at time t) such that expected number of cluster-head nodes for this round is k.

$$E[\#CH] = \sum_{i=1}^{N} P_i(t) * 1 = k. \qquad (1)$$

k= # of clusters during each round
N= # of nodes in the network

- Each node will be Cluster Head once in N/k rounds.
- Probability for each node i to be a cluster-head at time t

$$P_i(t) = \begin{cases} \frac{k}{N - k*(r \bmod \frac{N}{k})} & : \quad C_i(t) = 1 \\ 0 & : \quad C_i(t) = 0 \end{cases} \qquad (2)$$

$C_i(t)$ = it determines whether node i has been a cluster head in most recent *(r mod(N/k))* rounds.

CLUSTER HEAD SELECTION ALGORITHM

$$E\left[\sum_{i=1}^{N} C_i(t)\right] = N - k * \left(r \bmod \frac{N}{k}\right) \qquad (3)$$

$\sum_{i=1}^{N} C_i(t)$ = total no. of nodes eligible to be a **cluster-head** at time *t*.

This ensures energy at each node to be approx. equal after every N/k rounds.

Using (2) and (3), expected # of Cluster Heads per round is,

$$\begin{aligned} E[\#CH] &= \sum_{i=1}^{N} P_i(t) * 1 \\ &= \left(N - k * \left(r \bmod \frac{N}{k}\right)\right) * \frac{k}{N - k * \left(r \bmod \frac{N}{k}\right)} \\ &= k. \end{aligned} \qquad (5)$$

CLUSTER FORMATION ALGORITHM

Cluster Heads broadcasts an *advertisement message* (ADV) using *CSMA MAC protocol*.

- **ADV = node's ID + distinguishable header.**

Based on the *received signal strength* of ADV message, each non-Cluster Head node determines its Cluster Head for this round.

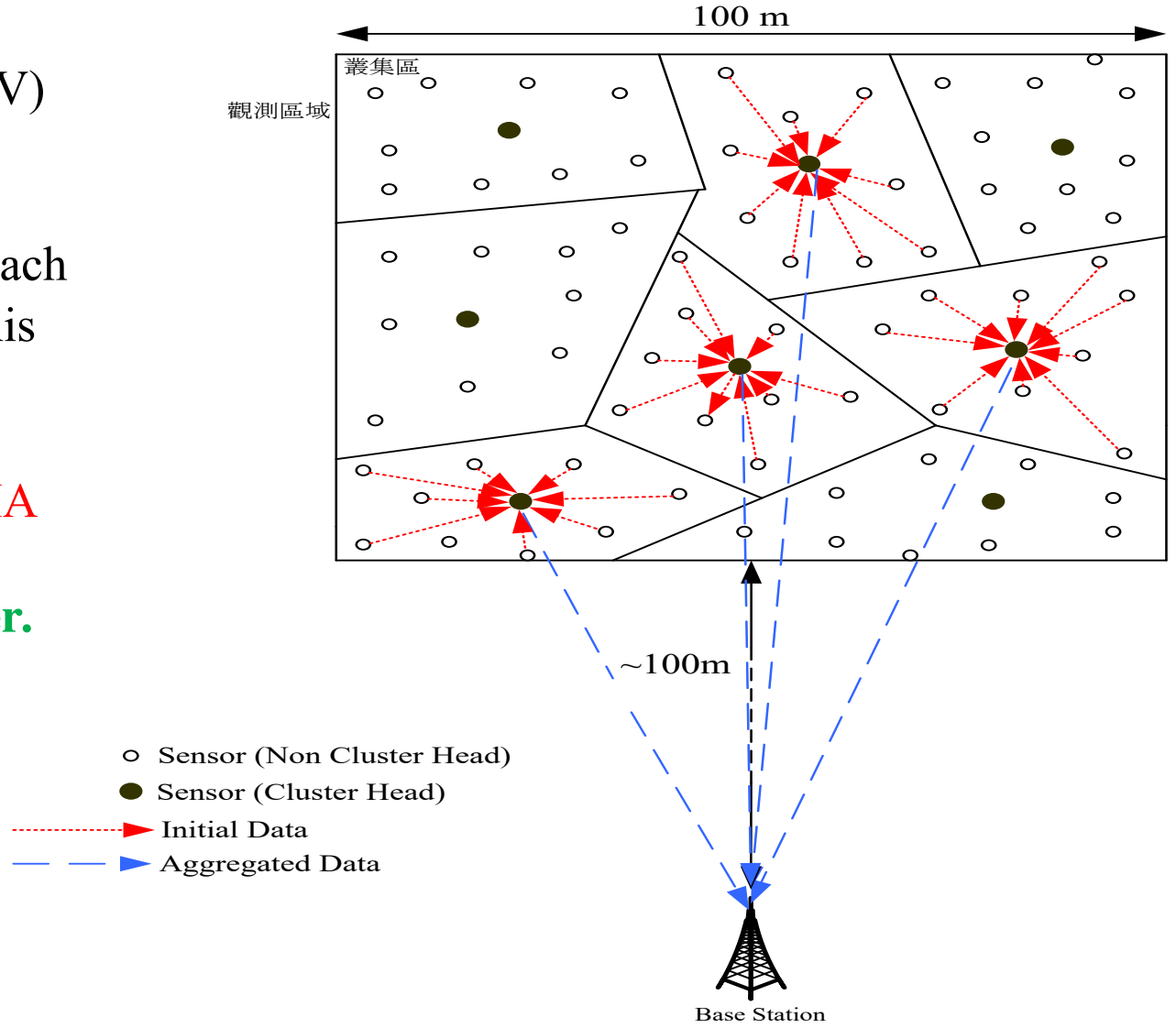
Each non-Cluster Head transmits a join-request message (Join-REQ) back to its chosen Cluster Head using a CSMA MAC protocol.

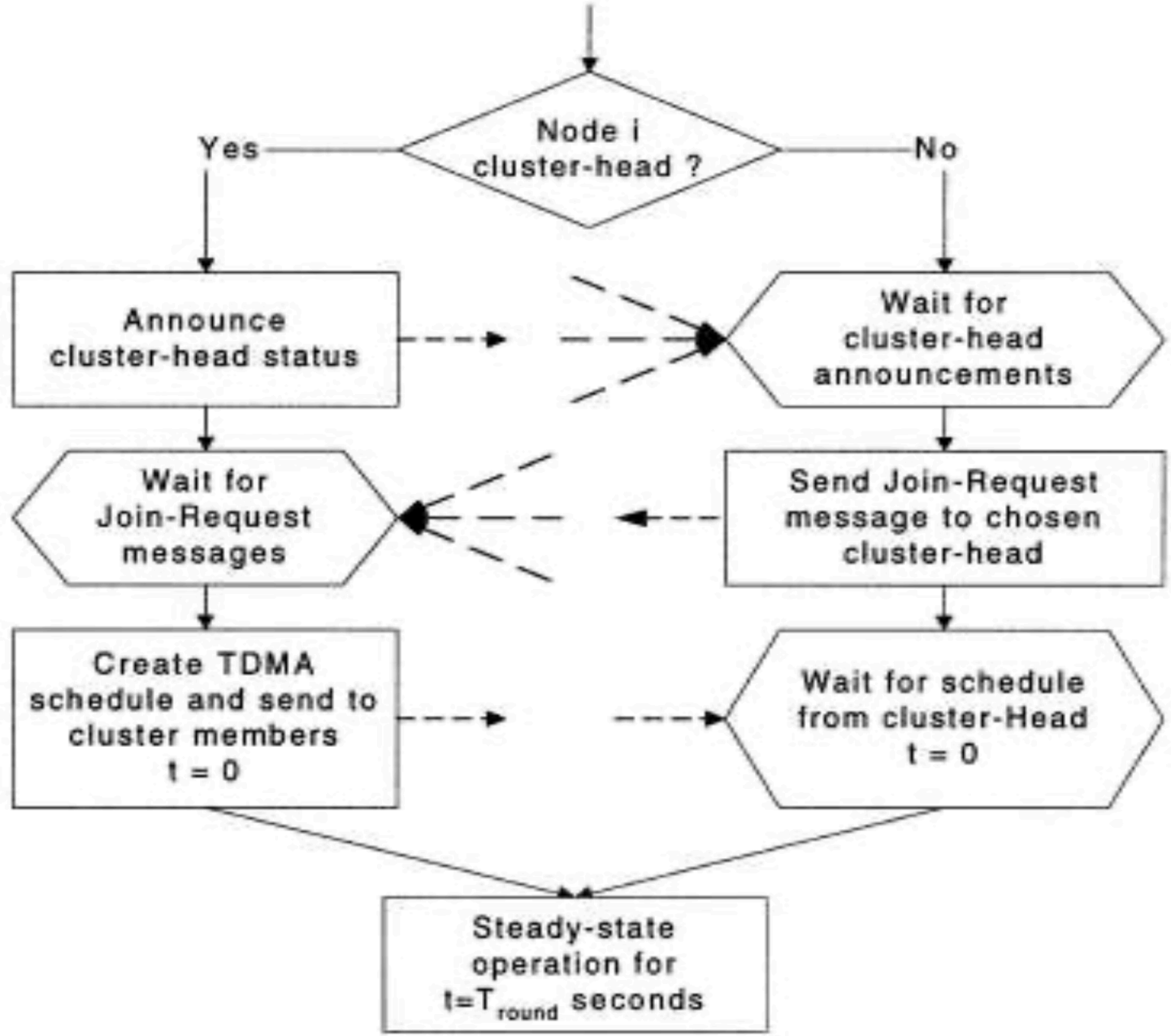
- **Join-REQ = node's ID + cluster-head ID + header.**

Cluster Head node sets up a TDMA schedule for data transmission coordination within the cluster.

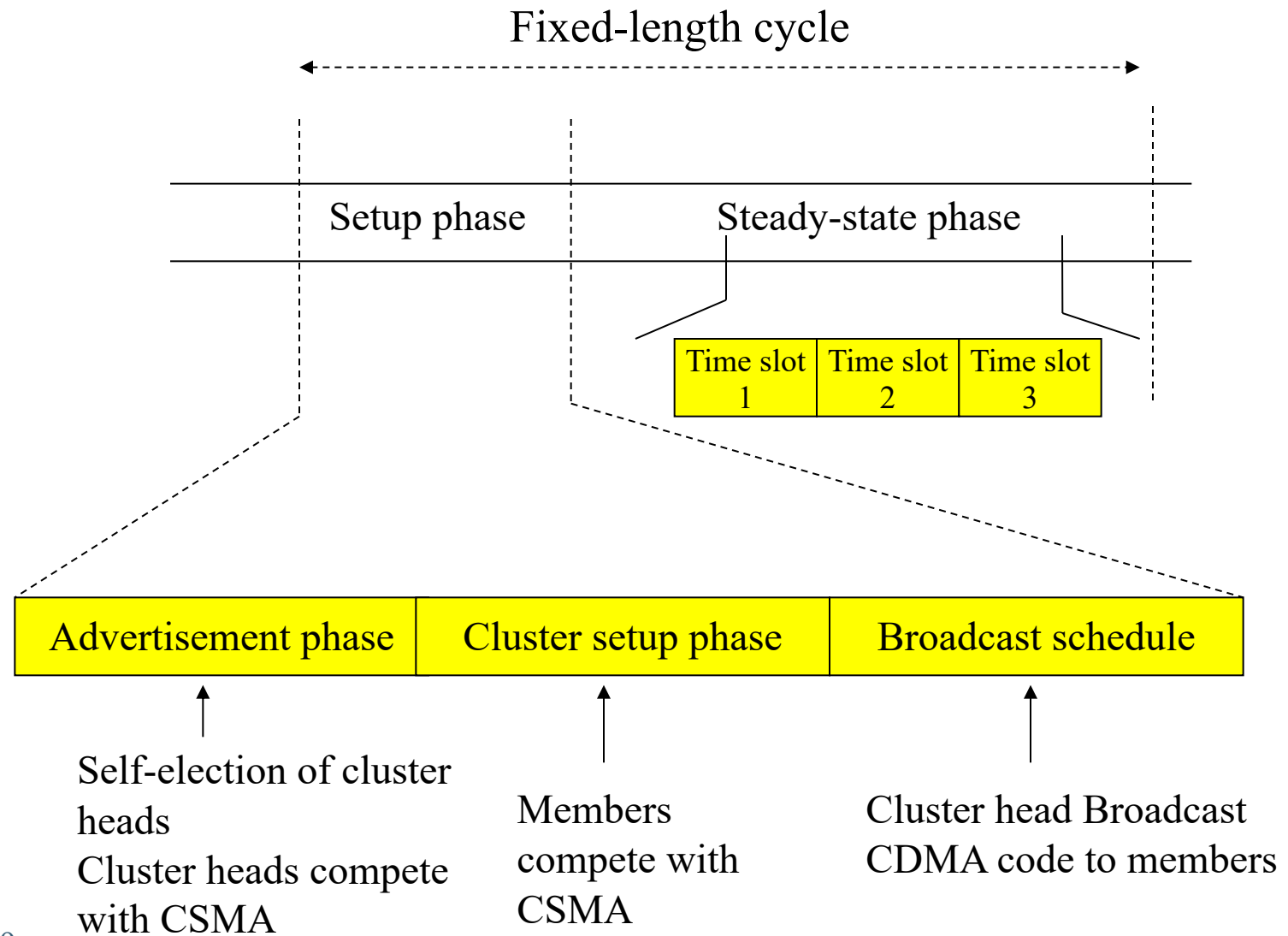
TDMA Schedule

- Prevents collision among data messages.
- Energy conservation in non cluster-head nodes.



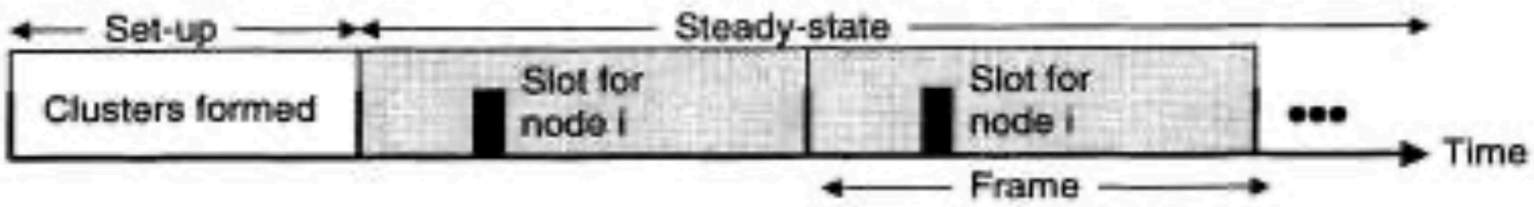


Flowchart of the distributed cluster formation algorithm for LEACH


Fixed-length cycle
Setup phase
Steady-state phase
Time slot 1
Time slot 2
Time slot 3
Advertisement phase
Cluster setup phase
Broadcast schedule
Self-election of cluster heads
Cluster heads compete with CSMA
Members compete with CSMA
Cluster head Broadcast CDMA code to members

9

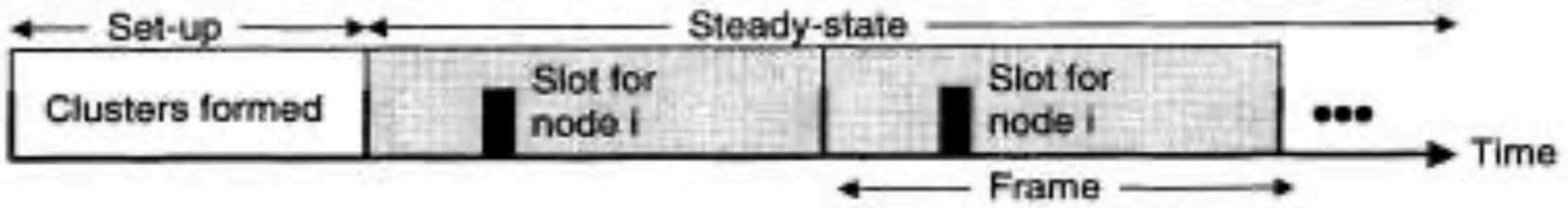
STEADY STATE PHASE



Time line showing LEACH operation

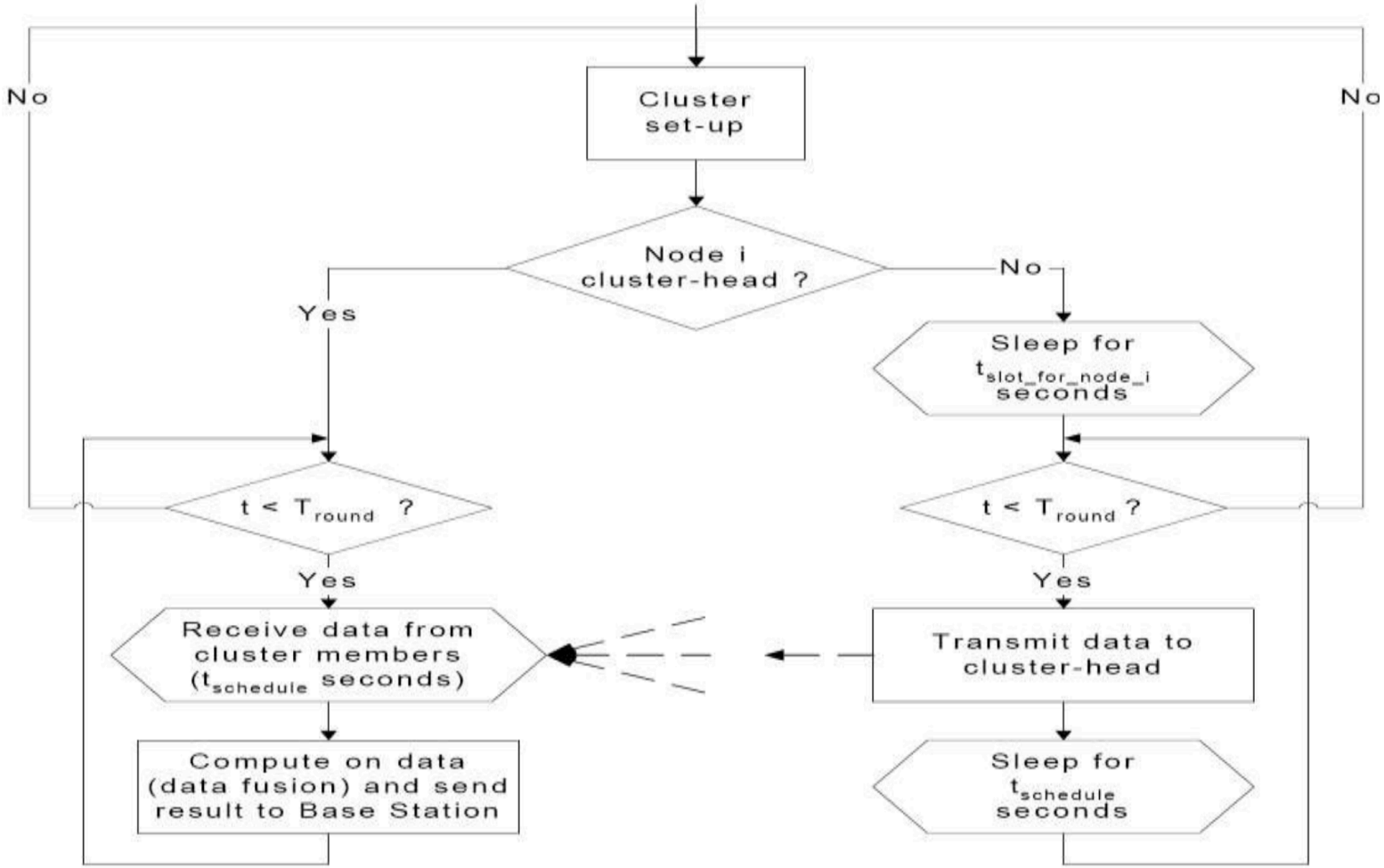
- TDMA schedule is used to send data from node to cluster head.
- Cluster head aggregates the data received from nodes in the cluster.
- Communication is via direct-sequence spread spectrum (DSSS) and each cluster uses a unique spreading code to reduce inter-cluster interference.
- Data is sent from the cluster head nodes to the BS using a fixed spreading code and CSMA.

STEADY STATE PHASE



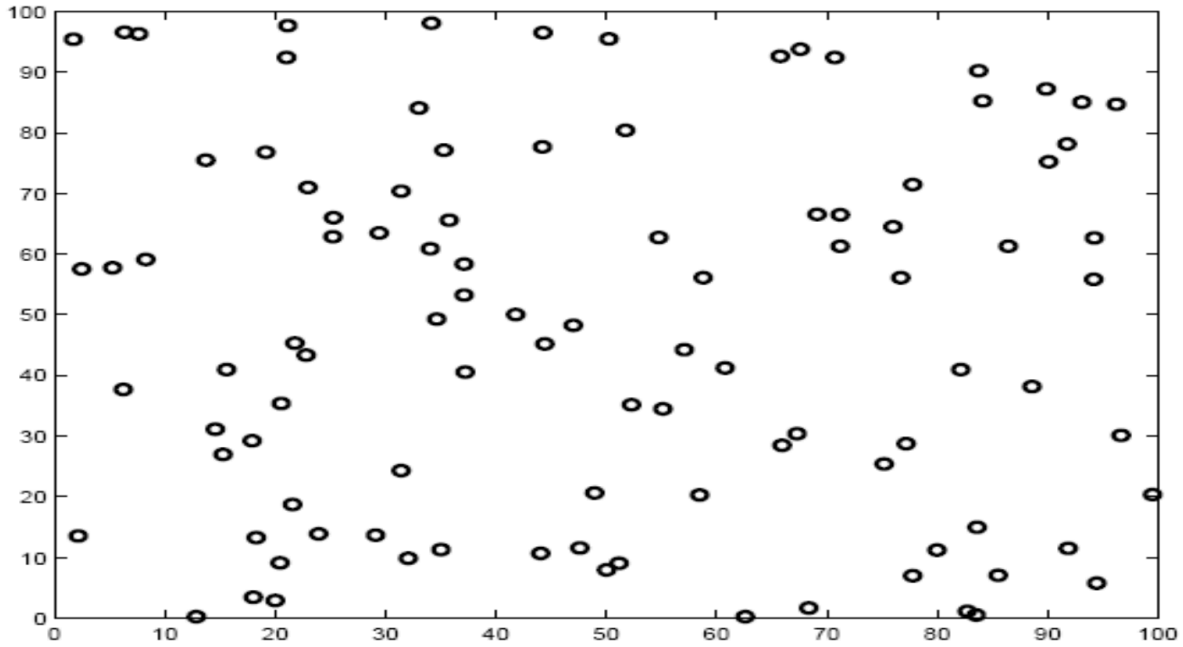
Time line showing LEACH operation

- Assumptions
  - ❖ Nodes are all time synchronized and start the setup phase at same time.
  - ❖ BS sends out synchronized pulses to the nodes.
  - ❖ Cluster Head must be awake all the time.
- To reduce inter-cluster interference, each cluster in LEACH communicates using direct-sequence spread spectrum (DSSS).
- Data is sent from the cluster head nodes to the BS using a fixed spreading code and CSMA.



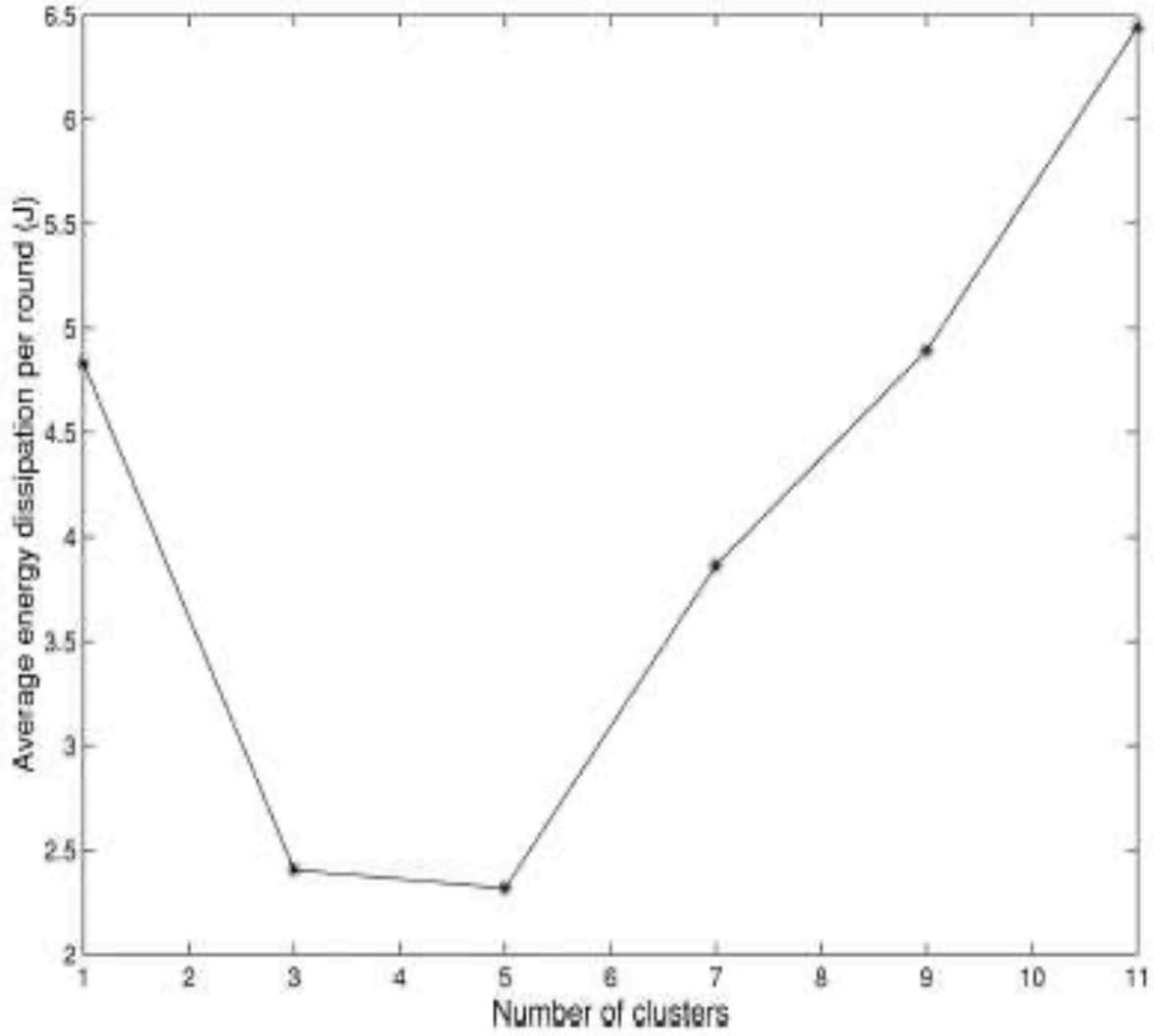
Flow chart for steady state phase

ANALYSIS AND SIMULATION

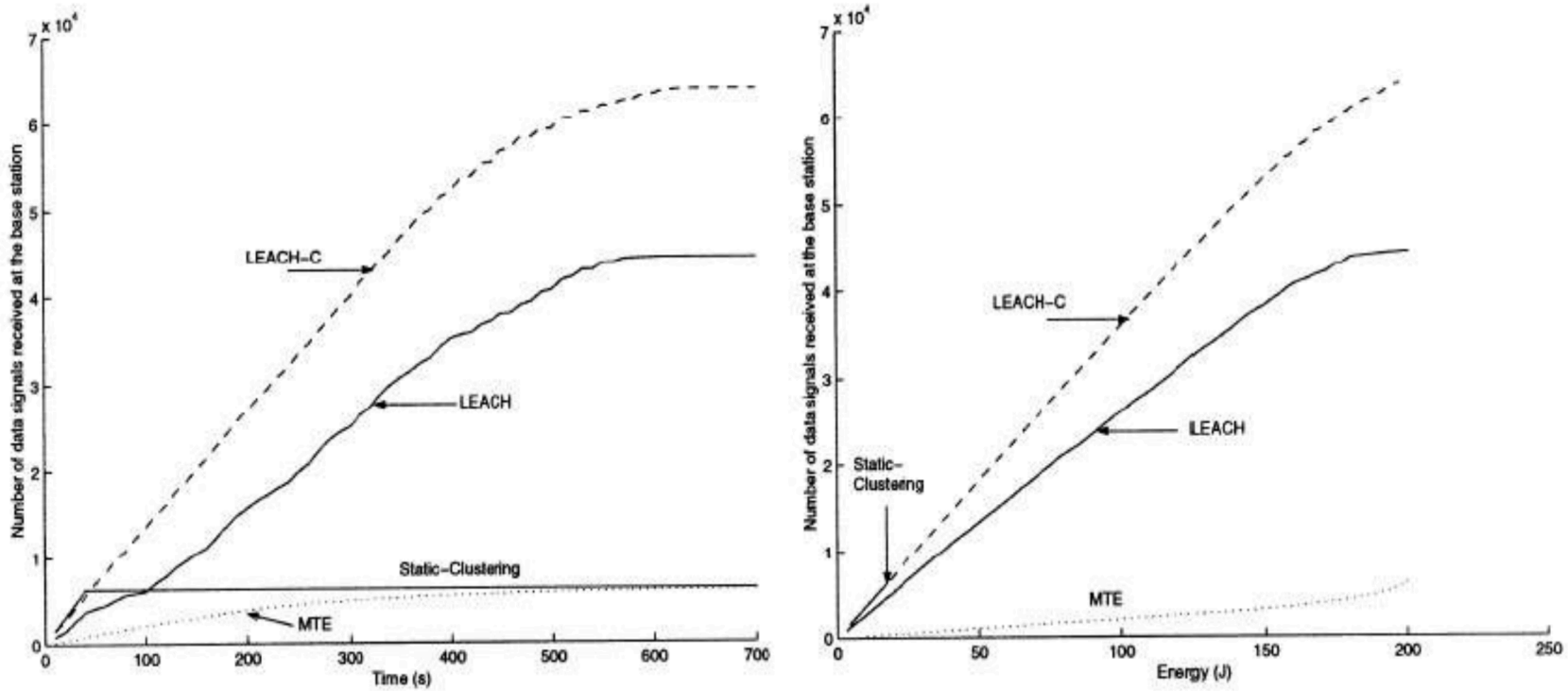


100 node random test network

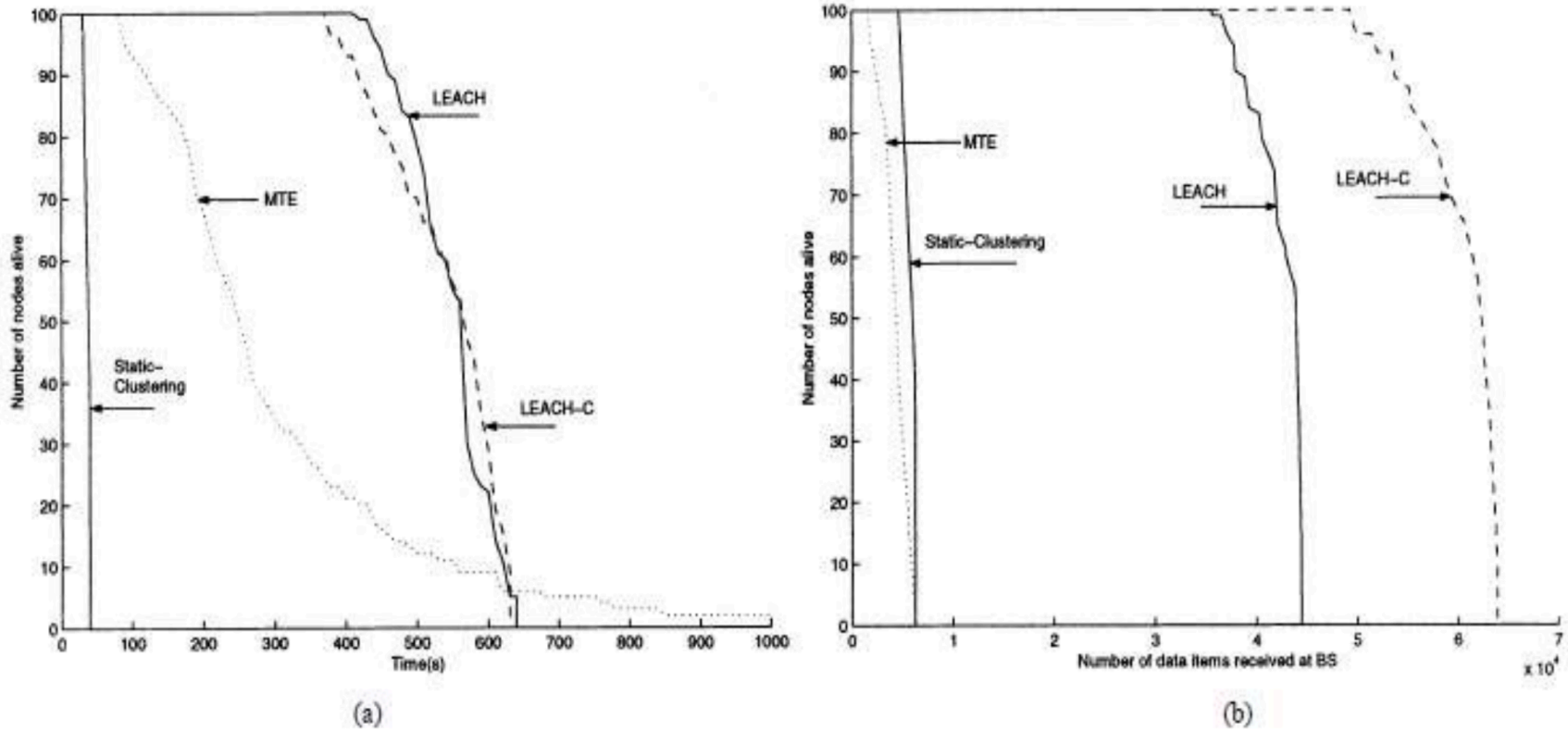
Nodes	100
Network size	100 m × 100 m
Base station location	(50, 175)
Radio propagation speed	$3x10^8$ m/s
Processing delay	50 μs
Radio speed	1 Mbps
Data size	500 bytes



Minimum Total Energy



- LEACH distributes more data per unit energy than MTE
- LEACH-C delivers 40% more data per unit energy than LEACH



- LEACH and LEACH-C's nodes lifetime is much longer than MTE's
- LEACH can deliver 10 times the amount of effective data to the BS for the same number of node deaths

Hierarchical vs. Flat Routing

Hierarchical routing	Flat routing
Reservation-based scheduling	Contention-based scheduling
Collisions avoided	Collision overhead present
Reduced duty cycle due to periodic sleeping	Variable duty cycle by controlling sleep time of nodes
Data aggregation by clusterhead	Node on multihop path aggregates incoming data from neighbors
Simple but non-optimal routing	Routing can be made optimal but with an added complexity.
Requires global and local synchronization	Links formed on the fly without synchronization
Overhead of cluster formation throughout the network	Routes formed only in regions that have data for transmission
Lower latency as multiple hops network formed by cluster- heads always available	Latency in waking up intermediate nodes and setting up the multipath
Energy dissipation is uniform	Energy dissipation depends on traffic patterns
Energy dissipation cannot be controlled	Energy dissipation adapts to traffic pattern
Fair channel allocation	Fairness not guaranteed

PERFORMANCE METRICS

Routing Overhead

- The routing overhead is the number of route packets sent with every data packets.
- It reflects the degree of network congestion and the efficiency of node power.
- Routing overhead can be expressed as:

$$N1 / N2$$

where $N1$ is the number of route packets which are sent and forwarded; $N2$ is the number of data packets received.

- The routing overhead is measured in bits per second (bit/s or bps).

PERFORMANCE METRICS

Average End to End Delay

- The average end to end delay includes all possible delays caused by buffering during route discovery latency, queuing at the interface queue, retransmission delays at the MAC and the propagation and transfer time.
- This metric is significant in understanding the delay introduced by path discovery. Average end to end delay can be expressed as:

$$SUM(T2-T1)/N$$

where *T2* is the time when the first data packet arrives to the destination *T1* is the time when the first packet is transmitted by source and *N* is the number of packets sent

PERFORMANCE METRICS

Throughput

- The throughput is the percentage of the packets received by the destination among the packets sent by the source.
- The throughput is given as:

$$N1/N2 * 100\%$$

where *N1* is the number of packets received and N2 is the number of packets sent and

- Throughput is measured in bps.

MACHINE LEARNING AND ROUTING PROTOCOLS

MACHINE LEARNING TECHNIQUES

Supervised Learning

- K-Nearest Neighbor (k-NN)
- Neural Networks (NNs)
- Support Vector Machines

Unsupervised Learning

- K-Means Clustering
- Principal Component Analysis (PCA)

Both Techniques require prior training data to learn best paths

In disaster scenario datasets are not available prior to incident
so we have to use online learning algorithm

CONT'D

In disaster scenario datasets are not available prior to incident so we have to use online learning algorithm

Received April 5, 2019, accepted April 23, 2019, date of publication April 29, 2019, date of current version May 9, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2913776

Reinforcement Learning Based Routing in Networks: Review and Classification of Approaches

ZOUBIR MAMMERI, **(Senior Member, IEEE)**
IRIT, Paul Sabatier University, 31062 Toulouse, France
e-mail: zoubir.mammeri@irit.fr

REINFORCEMENT LEARNING BAAED PROTOCOLS

In RL-based routing protocols, reward is addressed as a ‘cost’ to send packets.

From a networking point of view, cost may refer to delivery delay, loss rate, energy consumption, and so on.

The reward is calculated—at least partially—by a node upon selecting an entire route to use for all packets to transmit or just a next hop to transmit the current data packet.

Consequently, to fully comply with RL principles, a node should be considered to consist of an agent and optional components:

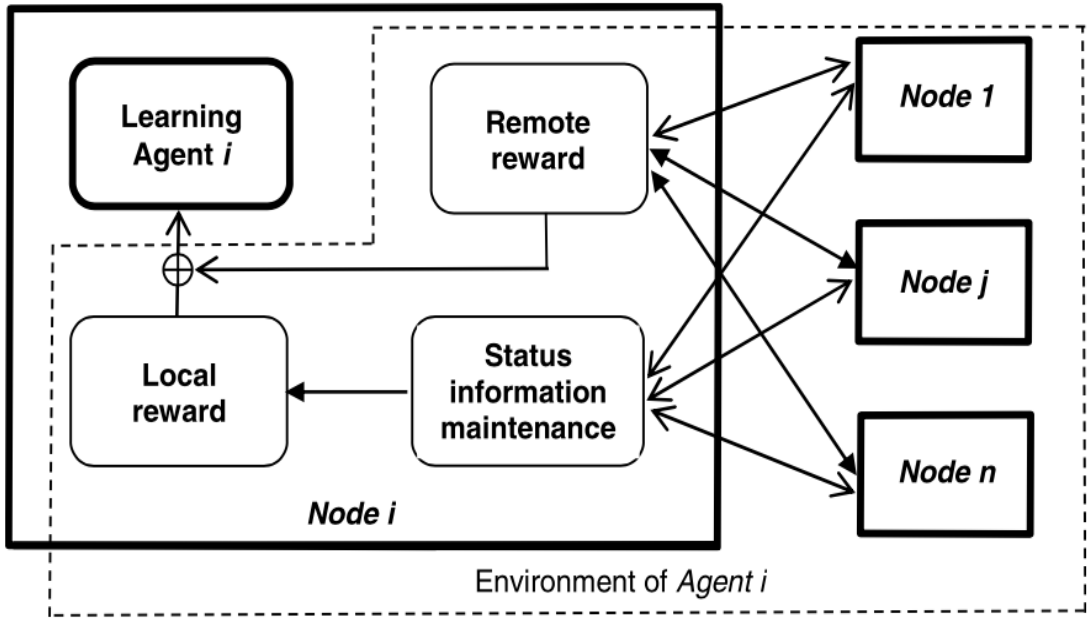


FIGURE 2. High-level structure of RL-based routing.

. This figure shows a high-level structure, which highlights components involved in RL-based routing, not all components are included in all existing routing protocols.

REINFORCEMENT LEARNING

Reinforcement Learning

- Q-Learning

Q-learning is a reinforcement learning algorithm.

The goal of Q-learning is to learn a policy, which tells an agent what action to take under what circumstances

AN EXAMPLE OF Q TAE

Initialized

Q-Table		Actions					
		South (0)	North (1)	East (2)	West (3)	Pickup (4)	Dropoff (5)
States	0	0	0	0	0	0	0
	.	.	.	.	.	.	.
	.	.	.	.	.	.	.
	.	.	.	.	.	.	.
	327	0	0	0	0	0	0
	.	.	.	.	.	.	.
	.	.	.	.	.	.	.
	.	.	.	.	.	.	.
	499	0	0	0	0	0	0

Training

Q-Table		Actions					
		South (0)	North (1)	East (2)	West (3)	Pickup (4)	Dropoff (5)
States	0	0	0	0	0	0	0
	.	.	.	.	.	.	.
	.	.	.	.	.	.	.
	.	.	.	.	.	.	.
	328	-2.30108105	-1.97092096	-2.30357004	-2.20591839	-10.3607344	-8.5583017
	.	.	.	.	.	.	.
	.	.	.	.	.	.	.
	.	.	.	.	.	.	.
	499	9.96984239	4.02706992	12.96022777	29	3.32877873	3.38230603

REL MODULES

- *Local reward* module, which calculates reward based on local view; local reward reflects the cost of communication as seen by packet sender.

- *Remote reward* module, which receives feedback sent by the next hop or by the destination node. Whenever both local and remote rewards are used in routing protocol, they are combined to form the reward returned to agent.

- *Link-state information maintenance* module, which collects useful link state information (such as the location and the residual energy of neighboring nodes and the quality of links) through periodic or on-demand *Hello beacon packets*.
- By shortcut, the latter are commonly called *Hello packets*. They are used for link state advertising in networks and consequently provide a support for collaboration of agents in RL-based routing algorithms.

OVERVIEW OF RL-BASED ROUTING PROTOCOLS

Boyan and Littman were the first to propose a hop-by-hop routing algorithm based on Q-learning, called Q-routing.

Many of existing RL-based routing protocols are extensions to Q-routing.

A. Q-ROUTING PROTOCOL

The name of the proposed algorithm comes from the notation of *Q-function* used in Q-learning method (§II.B). The algorithm may be summarized as follows:

Let *i*i denote the node holding a packet P to forward and $Q_i\,(d, j)$ denote the end-to-end delay (or simply delivery delay) that node *i*i estimates it takes, for node j, to deliver packet *P*i at destination *d*i. Node i maintains a table including the transfer delay estimates, called Q-values; a Q-value is associated with each neighbor of node i. When node i has a packet to send, it selects a node j with the lowest Q-value. Upon sending packet P to node j, node i immediately gets back j's an estimate for the time remaining in the trip to destination d denoted by $\theta_j(d)$:

$$\theta_j(d) = \min_{k \in Ng(j)} Q_j\,(d, k) \tag{9}$$

where $Ng(j)$ denotes the set of j's neighbors. Then node i updates its delivery delay estimate associated with neighbor j as follows:

$$Q_i\,(d, j) = (1 - \alpha) * Q_i\,(d, j) + \alpha * \left(qt_i + TxT_{i,j} + \theta_j(d)\right) \tag{10}$$

where α is a learning parameter, qt_i is the time spent by packet, P in i's queue and $TxT_{i,j}$ is the transmission time between nodes i and j. The pseudo code of Q-routing is the following:

To avoid frequent oscillations in Q-values (in case of sudden variations of traffic in the network) and limit the overhead of Q-routing protocol, [27] proposed an extension in which the receiver does not send an immediate reward for each received packet, but after receiving a certain number of packets (i.e., the receiver returns the average delay for a group of packets). However, Q-routing still suffers from at least the following:

- *Q-value freshness*. The estimate of delay occurs upon packet transmission on a route. When a route is not selected during a long period of time, the agent has no accurate estimate about the current condition of such a route and its Q-value may become unreliable. When a node resumes its transmission activity on a given route, after a long idling period, its delay estimate values may result in non-optimal selection of next hop.
- *Slow convergence*. Q-learning requires a number of epochs (or tries) before being able to converge to the optimal solution.
- *Parameter setting sensitivity*. Small adjustments in learning parameter may result in serious fluctuation in routing performance.

FACTORS FOR CALCULATING Q-VALUES

Residual Energy

Link Distance

Link Quality

Delivery time

Mobility

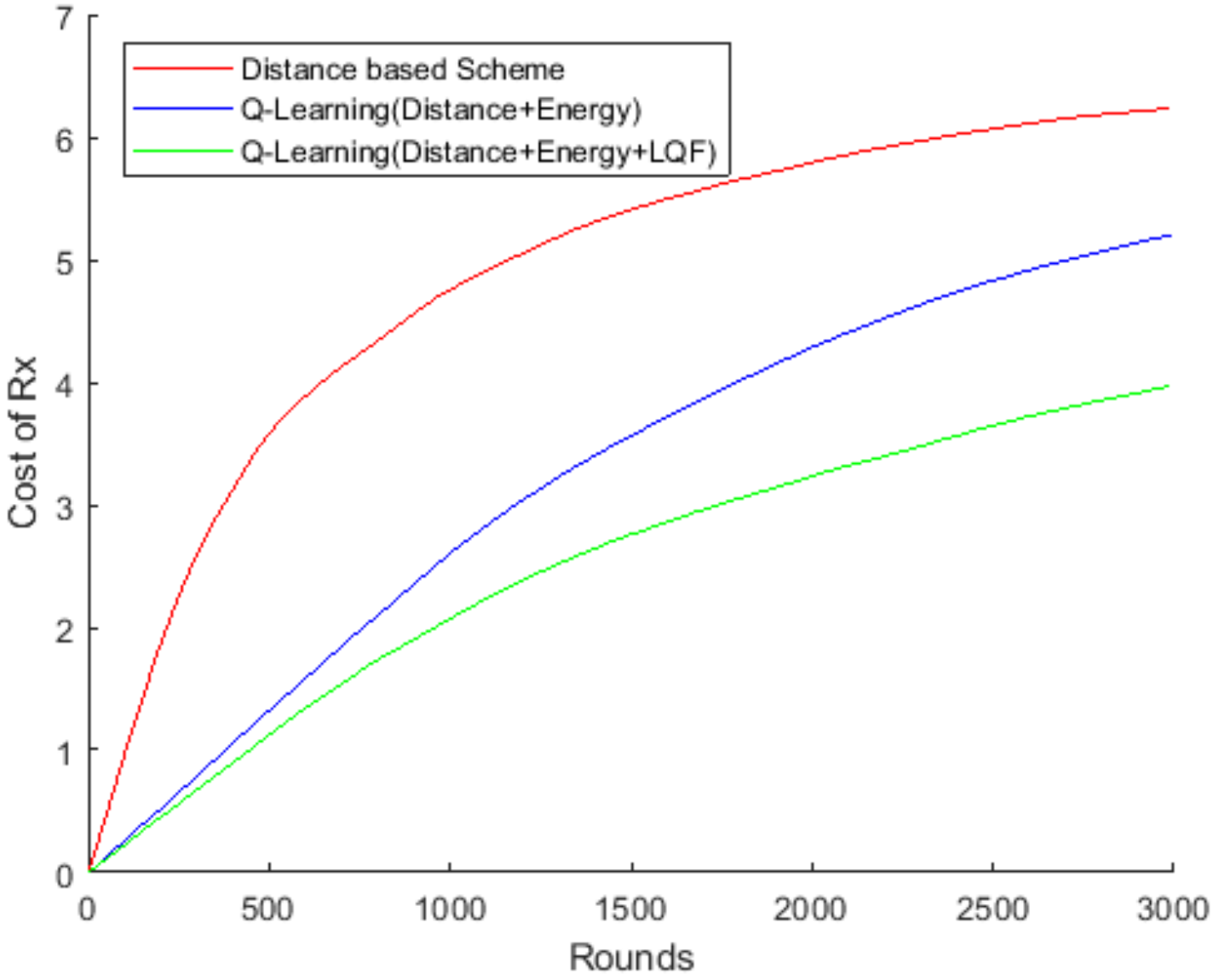
Hop Count

Position of Nodes

SIMULATION PARAMETER

Parameter	Value
Number of UE	100
Number of sink node	1
Topological area	100 m x 100 m
Deployment	Random
Transmission radius	25m
Initial energy	100%
Sink node deployment	(105,105)
Data packet size	1024 bytes
Learning rate	0.5

SIMULATION RESULTS



CLUSTERING USING Q-VALUES

Nodes with max Q-Values are selected as cluster heads

Nodes that falls in the overlapping region of two or more clusters are declared as gateways. They are used for communication between the adjacent clusters.

SIMULATION SCENARIO