DDR-Distributed Dynamic Routing Algorithm for Mobile Ad Hoc Networks

Navid NIKAEIN
Houda LABIOD
Christian BONNET

EURECOM Institute
Sophia-Antipolis

France

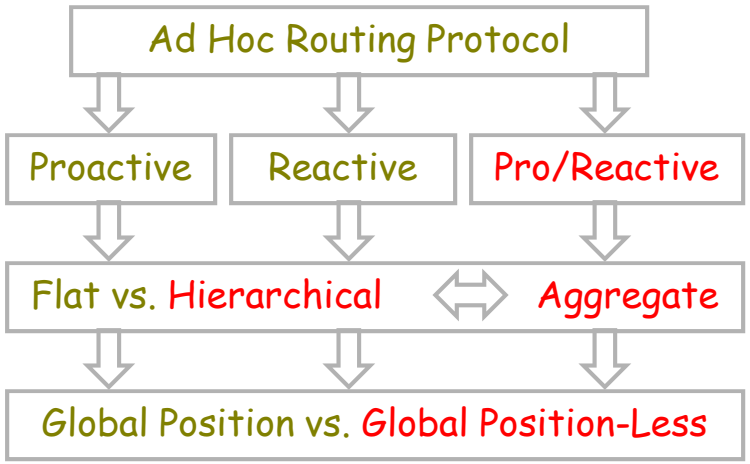
EURECOM

Contents

- Introduction
- Basic Idea
- DDR - Distributed Dynamic Routing Algorithm Description
- Conclusion and Future Work

Introduction

A mobile ad hoc network is a set of wireless nodes forming dynamic autonomous networks.

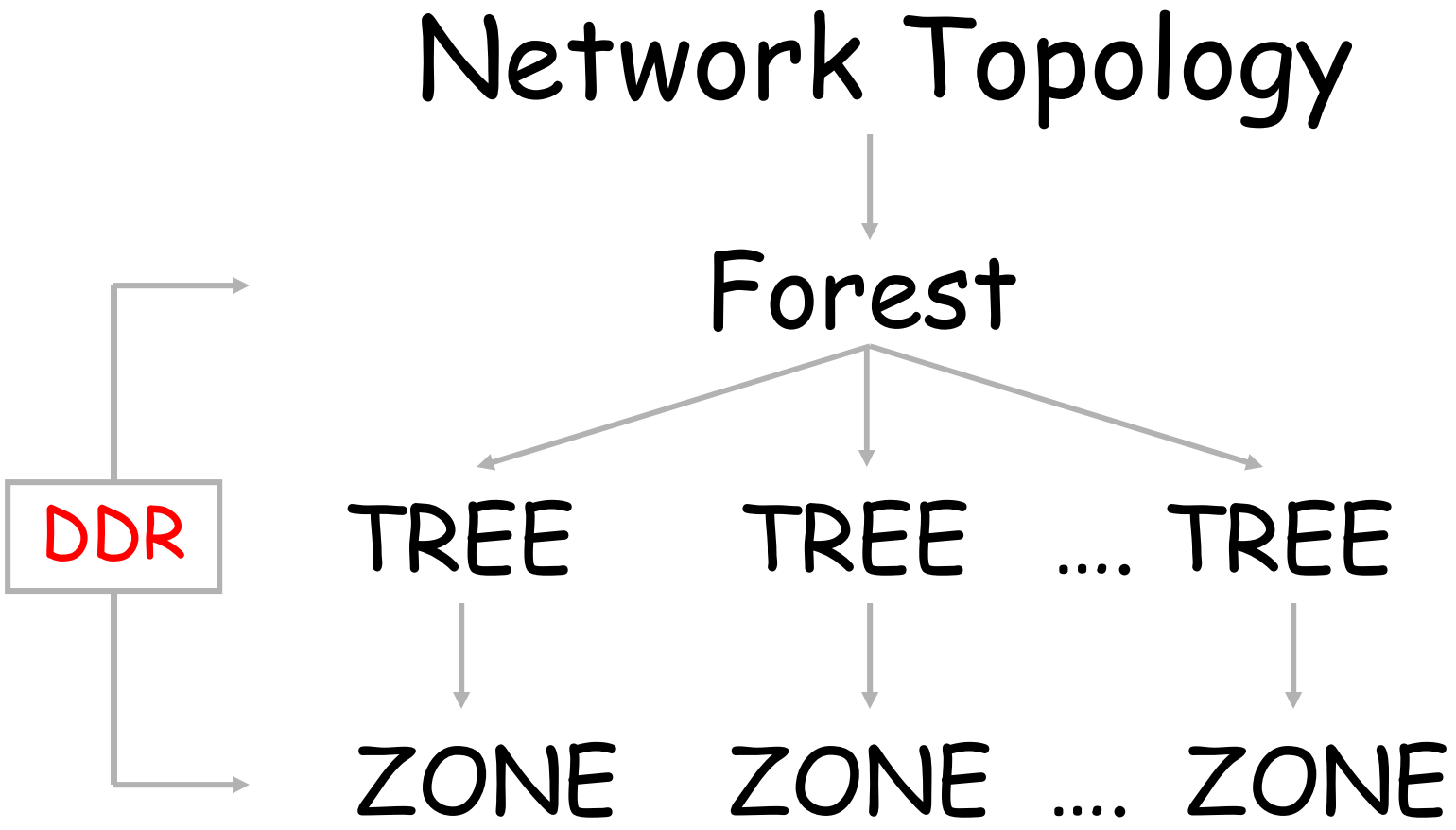


Routing

Critical key features of routing protocols:

- Optimal Path:
  - #hops, most stable, ↓delay, ↓energy and ↓loss rate to Dest,
- Fast adaptability to link changes,
- Distributed operation,
- Loop avoidance.

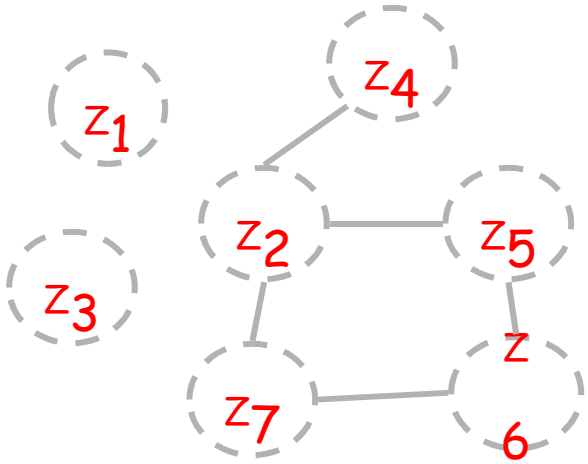
Basic Idea (1)

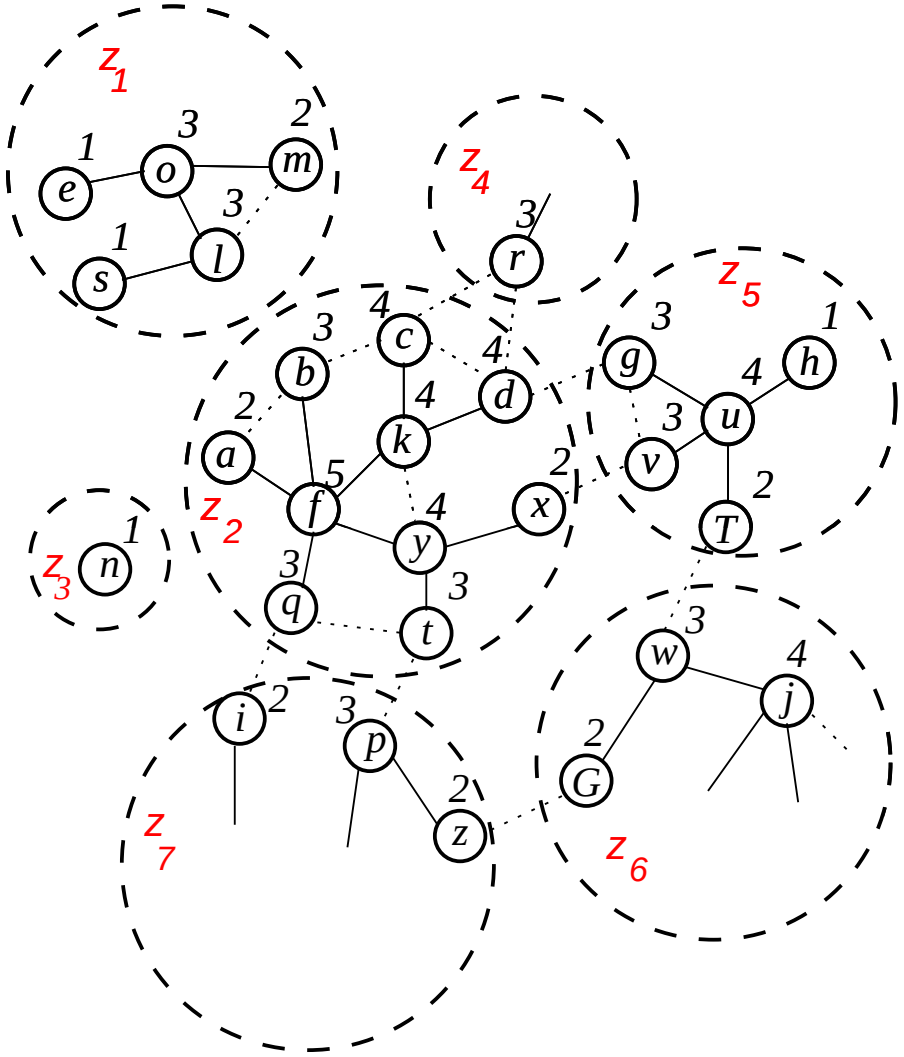


Basic Idea (2)

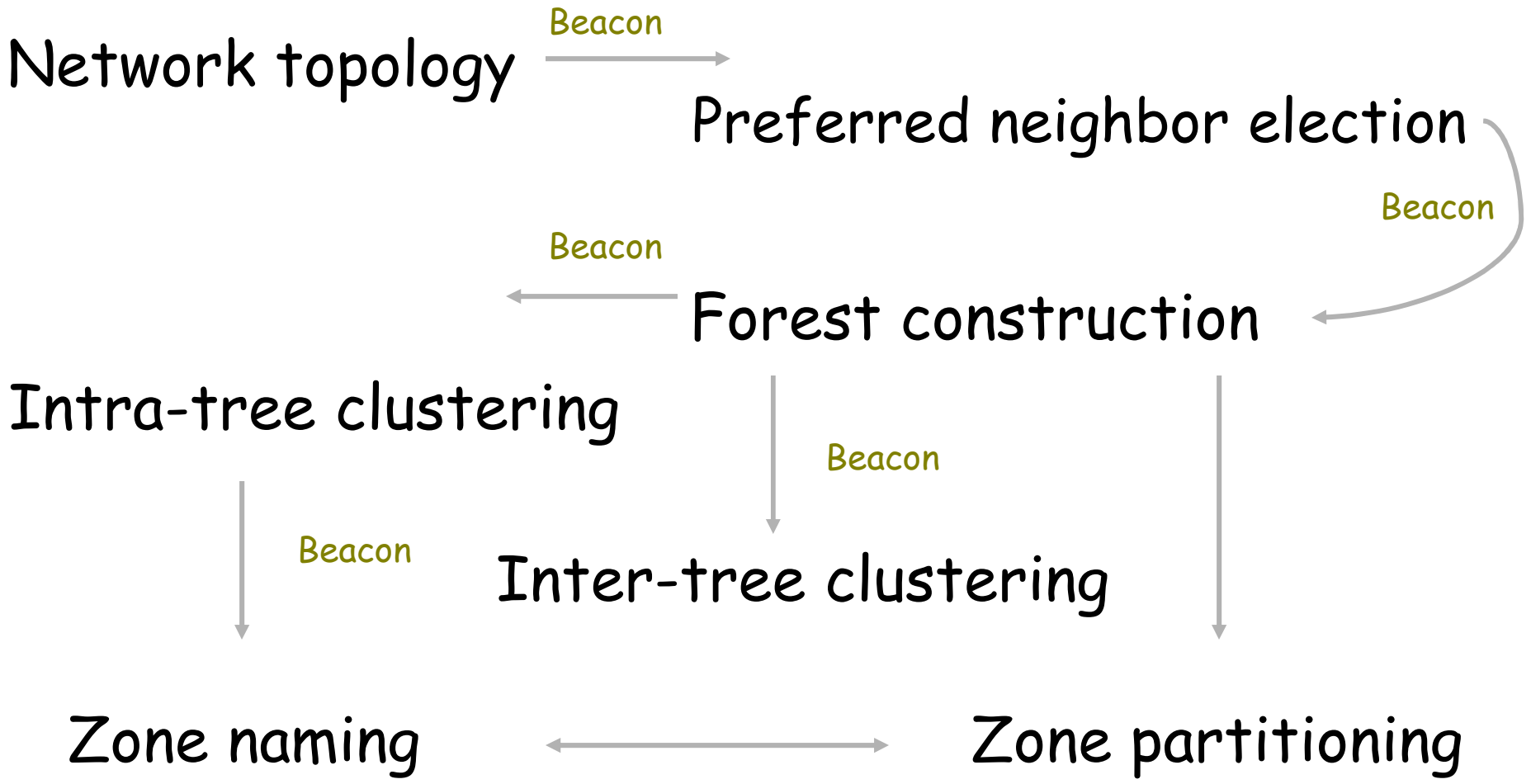
❶ Forest partitions the network into a set of non over-lapping dynamic zones.

Reduced graph: $G'=(V', E') \leq G=(V, E)$





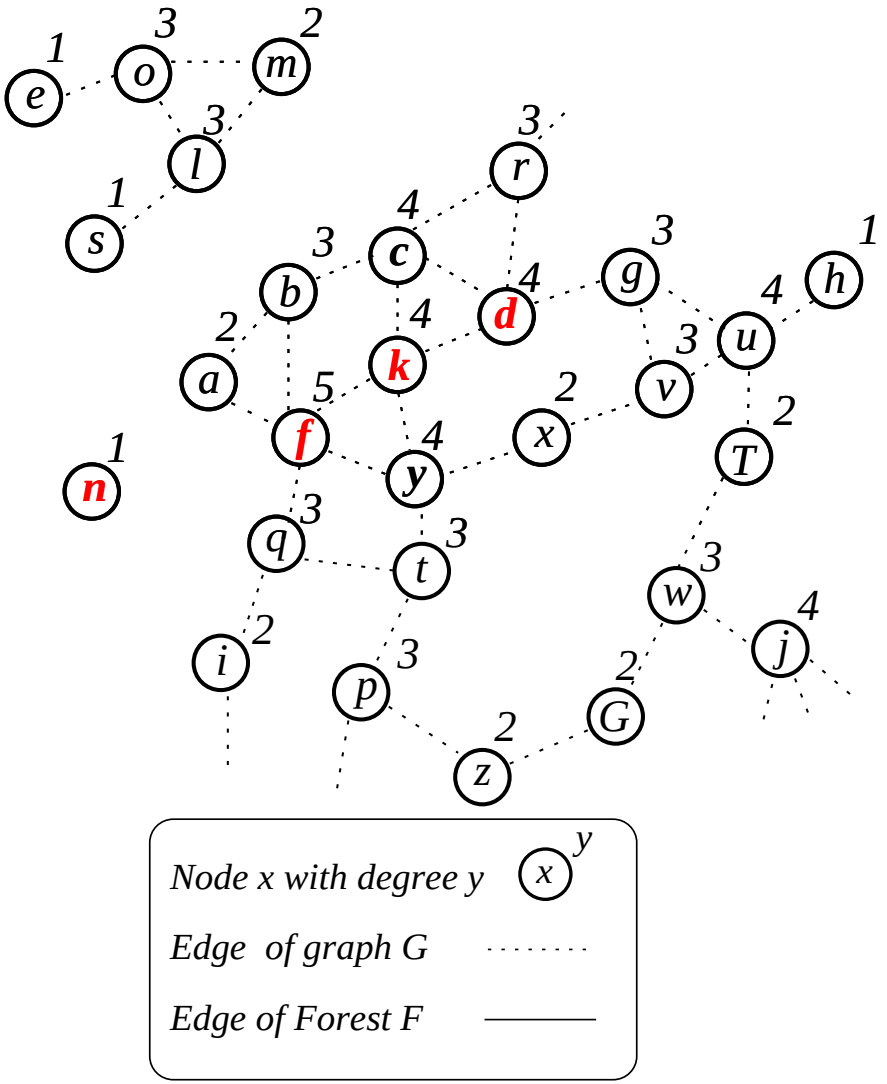
DDR - Algorithm



Preferred Neighbor Election

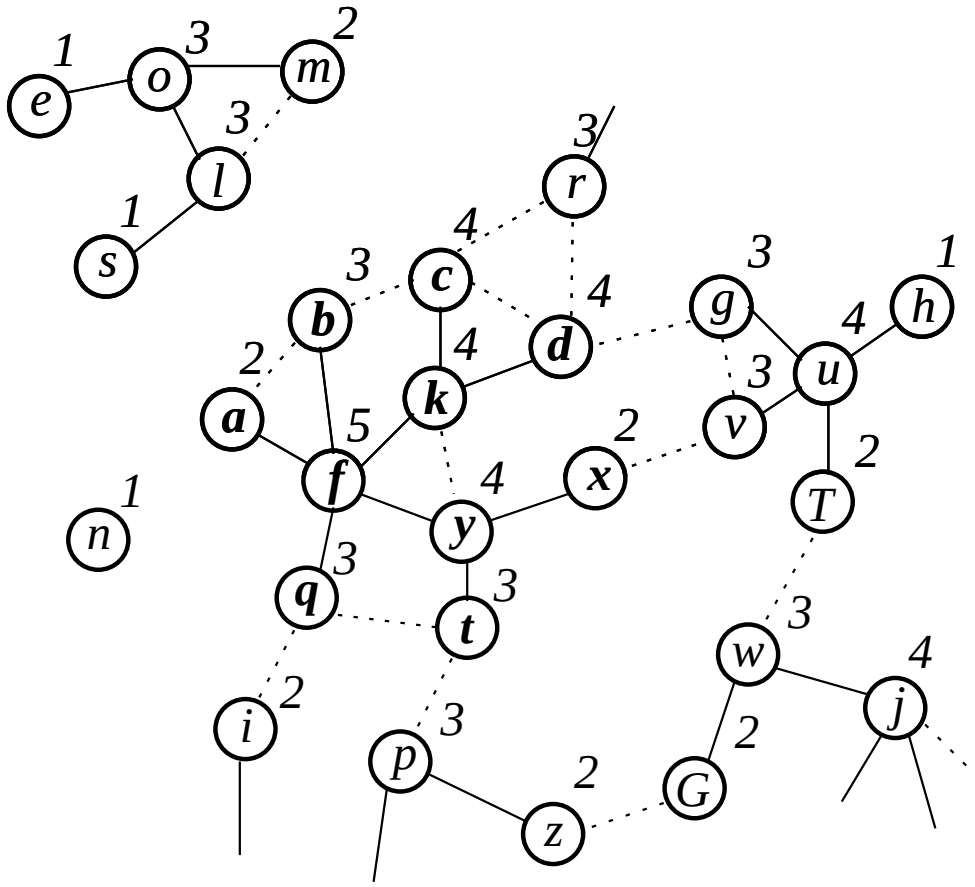
$PN_X = \{y \mid deg(y) = Max(deg(N_X))\}$

1. If this set is empty, then no PN, e.g. $PN_n = \{\emptyset\}$.
2. If this set has only one member, then this member is the elected PN, e.g. $PN_k = \{f\}$.
3. If more than one member has *max* degree, then we select the one with *max* ID number, e.g. $PN_d = \{c, K\}$.



Forest Construction

- A forest is constructed by connecting each node to its preferred neighbor and vice versa.



Intra-tree Clustering (1)

Node k:

❶ PN = f then B = (ZID, k, 4, 1, f)

❷ Learned_PN = c: d

❸ if (PN not changed) then

B = (ZID, k, 4, 0, c: d)

❹ Learned_PN = a: b: q: y

Node f:

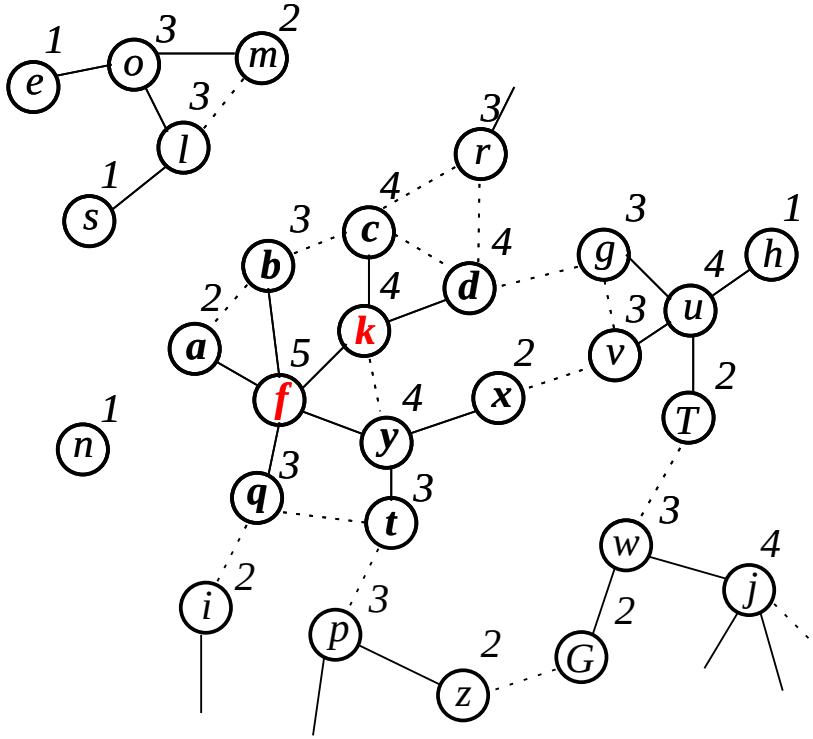
❶ PN = y then B = (ZID, f, 5, 1, y)

❷ Learned_PN = a: b: q: y: k

❸ if (PN not changed) then

B = (ZID, f, 5, 0, a: b: q: y: k)

❹ Learned_PN = c: d: x: t



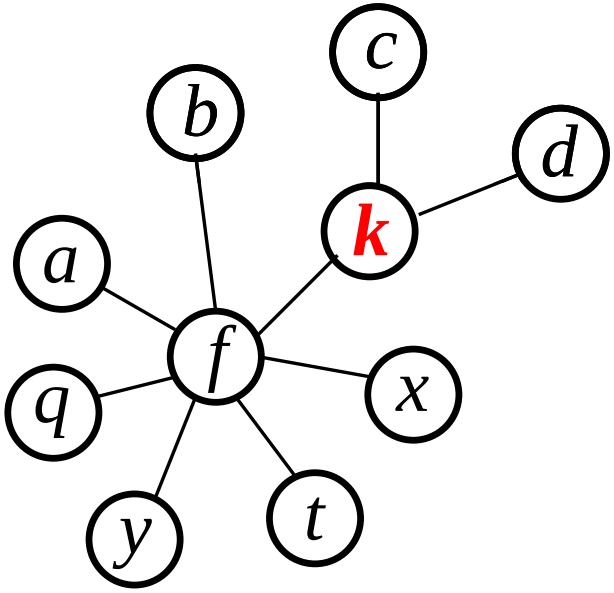
INTRA-ZONE TABLE OF NODES *k* AND *f*

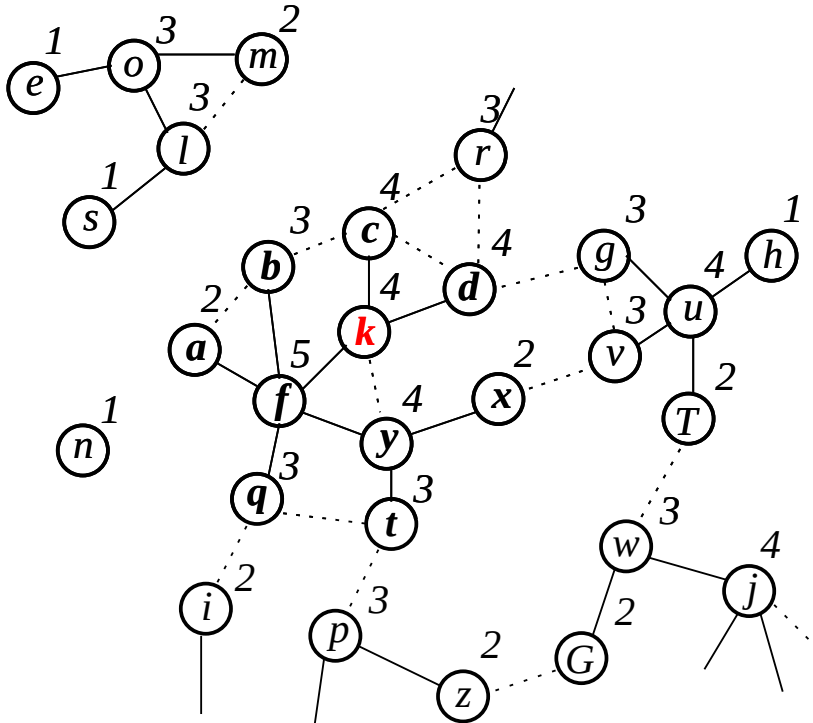
NID	Learned_PN
f	a, b, q, y, t, x
c	-
d	-

NID	Learned_PN
y	x, t
k	c, d
b, a, q	-

Intra-tree Clustering (2)

VIEW OF NODE K
ABOUT ITS TREE





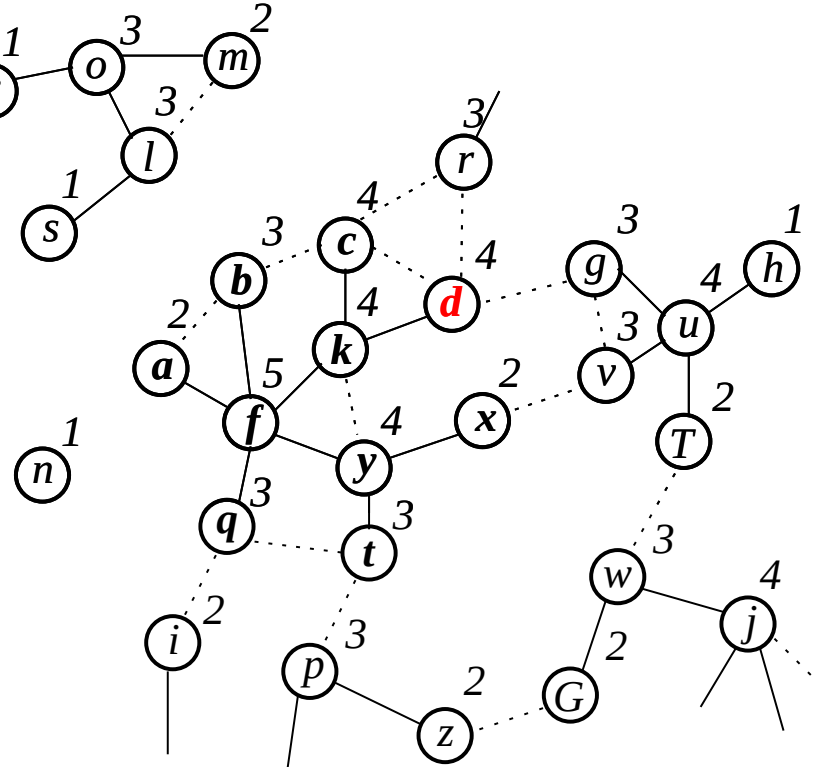
INTRA-ZONE TABLE OF NODES *k* AND *f*

NID	Learned_PN
f	a, b, q, y, t, x
c	-
d	-

NID	Learned_PN
y	x, d
k	c, d
b, a, q	-

Inter-tree Clustering

1. Either a node can succeed to add some nodes to its intra-zone table.
2. Otherwise, it puts the remaining nodes in its inter-zone table.

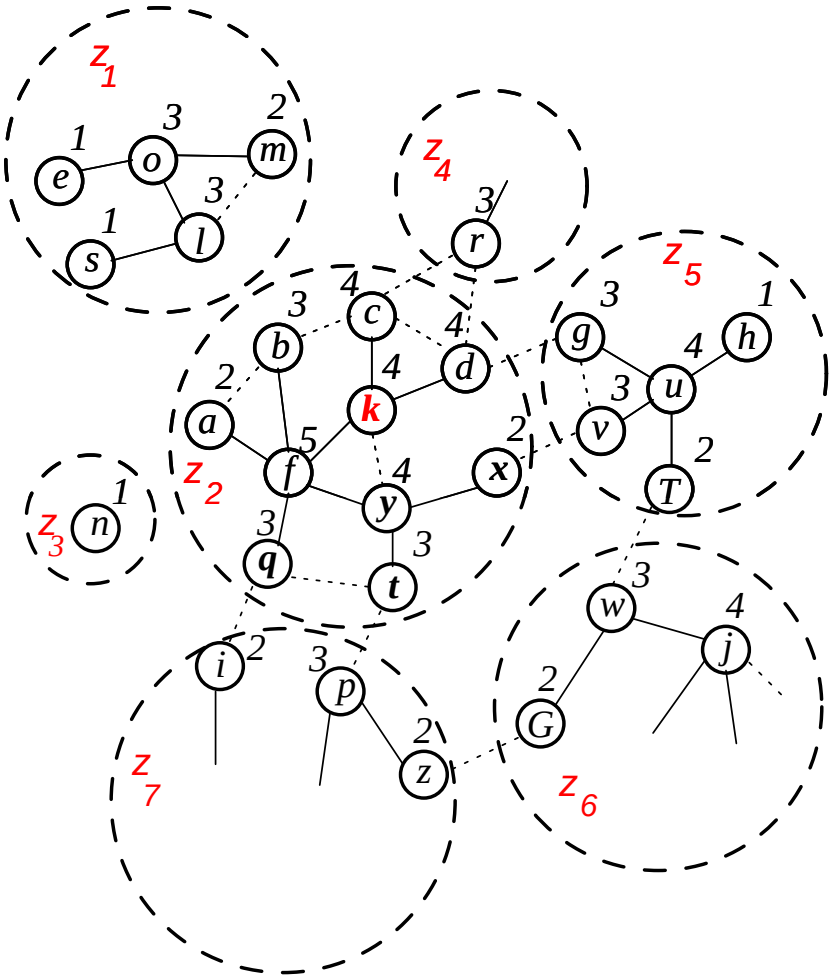


INTER-ZONE TABLE OF NODE *d*

NID	NZID	Z_Stability
r	z4	++
g	z5	++

Zone Naming

1. Select q highest ID # in intra-zone table, where $q = \left\lfloor \frac{n}{d} \right\rfloor$.
2. Compute a hash function on each selected ID # separately.
3. Concatenate all the hashed ID #.



Node k (for n=12 & d=4) :

1. q = 4 ⇨ selected nodes: y, x, t, q
2. h(y)|h(x)|h(t)|h(q)
3. Z_2= y'x't'q'

Conclusion and Future Work

- DDR algorithm is:
  - Simple,
  - Loop-free,
  - Distributed,
  - Bandwidth-efficient.
- Routing protocol description with both numerical and performance analysis.
- Mobile agent over ad hoc networks.

Preliminary Definitions

- Beacon

ZID	NID	NID_DEG	MY_PN	PN

- Intra-zone table

NID	Learned_PN

- Inter-zone table

GNID	NZID	Z_Stability