DDR-Distributed Dynamic Routing Algorithm for Mobile Ad Hoc Networks

Navid NIKAEIN
Houda LABIOD
Christian BONNET

EURECOM Institute
Sophia-Antipolis
France
Introduction, Basic Idea, DDR-Algorithm, Conclusion and Future Work
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)

Network Topology

- Forest
  - TREE
  - TREE
  - TREE
  - ZONE
  - ZONE
  - ZONE

Introduction, Basic Idea, DDR-Algorithm, Conclusion
Basic Idea (2)

1. Forest partitions the network into a set of non-overlapping dynamic zones.

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

Introduction, Basic Idea, DDR-Algorithm, Conclusion
DDR - Algorithm

- Network topology
- Preferred neighbor election
- Forest construction
- Intra-tree clustering
- Inter-tree clustering
- Zone naming
- Zone partitioning

Introduction, Basic Idea, DDR-Algorithm, Conclusion
Preferred Neighbor Election

\[ PN_x = \{ y \mid \text{deg}(y) = \text{Max}(\text{deg}(N_x)) \} \]

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. \( PN_d = \{ c, K \} \).

Introduction, Basic Idea, DDR-Algorithm, Conclusion
Forest Construction

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

Introduction, Basic Idea, DDR-Algorithm, Conclusion
Intra-tree Clustering (1)

Node k:
1. PN = f then B = (ZID, k, 4, 1, f)
2. Learned_PN = c: d
3. if (PN not changed) then
   B = (ZID, k, 4, 0, c: d)
4. Learned_PN = a: b: q: y

Node f:
1. PN = y then B = (ZID, f, 5, 1, y)
2. Learned_PN = a: b: q: y: k
3. if (PN not changed) then
   B = (ZID, f, 5, 0, a: b: q: y: k)
4. Learned_PN = c: d: x: t

INTRA-ZONE TABLE OF NODES k AND f

<table>
<thead>
<tr>
<th>NID</th>
<th>Learned_PN</th>
</tr>
</thead>
<tbody>
<tr>
<td>f</td>
<td>a, b, q, y, t, x</td>
</tr>
<tr>
<td>c</td>
<td>-</td>
</tr>
<tr>
<td>d</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NID</th>
<th>Learned_PN</th>
</tr>
</thead>
<tbody>
<tr>
<td>y</td>
<td>x, t</td>
</tr>
<tr>
<td>k</td>
<td>c, d</td>
</tr>
<tr>
<td>b, a, q</td>
<td>-</td>
</tr>
</tbody>
</table>

Introduction, Basic Idea, DDR-Algorithm, Conclusion
Intra-tree Clustering (2)

VIEW OF NODE K ABOUT ITS TREE

Introduction, Basic Idea, DDR-Algorithm, Conclusion
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.

<table>
<thead>
<tr>
<th>NID</th>
<th>NZID</th>
<th>Z_ Stability</th>
</tr>
</thead>
<tbody>
<tr>
<td>r</td>
<td>z4</td>
<td>++</td>
</tr>
<tr>
<td>g</td>
<td>z5</td>
<td>++</td>
</tr>
</tbody>
</table>

Introduction, Basic Idea, DDR-Algorithm, Conclusion
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. $q = 4 \Rightarrow$ 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.

Introduction, Basic Idea, DDR-Algorithm, Conclusion
Preliminary Definitions

- **Beacon**
  - `ZID` | `NID` | `NID_DEG` | `MY_PN` | `PN`

- **Intra-zone table**
  - `NID` | `Learned_PN`

- **Inter-zone table**
  - `GNID` | `NZID` | `Z_Stability`

Introduction, Basic Idea, **DDR-Algorithm**, Conclusion