Classification of Ad Hoc Routing Protocols

Prasannakumar. M.
M.tech 1st year,CSE
S J B Institute of Technology
Bangalore,India
prasifrnd4evr@gmail.com

Mrs. Rekha.V
Asso.Prof.
S J B Institute of Technology
Bangalore,India

***Abstract*:Mobile ad hoc networks (MANET) are networks which routing is based on multi-hop routing from a source to a destination node or nodes. These networks have quite a many constrains because of uncertainty of radio interface and its limitations e.g. in available bandwidth. Also some terminals have limitations concerning battery energy in use.**

There are numerous applicable protocols for ad hoc networks, but one confusing problem is the vast number of separate protocols. Each of these protocols is designed to perform its task as well as it is possible according to its design criteria. The protocol to be chosen must cover all states of a specified network and never is allowed to consume too much network resources by protocol overhead traffic.

This seminar paper deals with a classification of ad hoc routing protocols and also presents some specified protocols according to that classification. Presented protocols are selected according to an entity formed by this paper and related papers to be published by Networking Laboratory of HUT. The emphasis of this paper is not to present protocols in detail but to present main features of wide variety of different protocols and evaluate their suitability and tradeoffs.

I. INTRODUCTION

Ad hoc network is a multi-hop wireless network, which consists of number of mobile nodes. These nodes generate traffic to be forwarded to some other nodes or agroup of nodes. Due to a dynamic nature of ad hoc networks, traditional fixed network routing protocols are not viable. Based on that reason several proposals for routing protocols has been presented.

Ad hoc radio networks have various implementation areas. Some areas to be mentioned are military, emergency, conferencing and sensor applications. Each of these application areas has their specific requirements for routing protocols. For example in military applications low probability of detection and interceptionis a key factor such is routing efficiency during fading and disturbed radio channel conditions.

At sensor applications low or minimum energy consumption is a precondition for an autonomous operation. In conference applications a guaranteed quality of service for multimedia services is a needed feature.

All application areas have some features and requirements for protocols in common. The routing protocol overhead traffic is not allowed to drive the network to congestion nor is a local change in link not allowed to cause a massive control traffic storm throughout the network.

II. A TAXONOMY FOR ROUTING PROTOCOLS

Because of multiple and diverse ad hoc protocols there is an obvious need for a general taxonomy to classify protocols considered. Traditional classification is to divide protocols to table-driven and to source-initiated on-demand driven protocols .

Table-driven routing protocols try to maintain consistent, up-to-date routing information from each node to every other node. Network nodes maintain one or many tables for routing information. Nodes respond to network topology changes by propagating route updates throughout the network to maintain a consistent network view.

Source-initiated on-demand protocols create routes only when these routes are needed. The need is initiated bythe source, as the name suggests. When a node requires aroute to a destination, it initiates a route discovery process within the network. This process is completedonce a route is found or all possible route permutationshave been examined. After that there is a routemaintenance procedure to keep up the valid routes and to remove the invalid routes.

This classification has though some drawbacks because of its rough granularity. To that classification it ispossible to make some modifications (e.g. in [2]). Thesemodifications can make some assumption about if the routing is flat or hierarchical and if any means to obtainglobal positioning information is in use.

One very attractive taxonomy has been introduced by Feeney [3]. This taxonomy is based on to divide protocols according to following criteria, reflectingfundamental design and implementation choices:

- **Communication model.** What is the wireless communication model? Multi-or single- channel?

- **Structure.** Are all nodes treated uniformly? How are distinguished nodes selected? Is theaddressing hierarchical or flat?
- **State Information.** Is network-scale topology information obtained at each node?
- **Scheduling.** Is route information continuallymaintained for each destination?

This model does not take an account for if a protocol isunicast, multicast,geocast or broadcast. Also the taxonomy doesn't deal with the question how the link ornode related costs are measured. These properties arehowever worth to be considered in classification and evaluating applicability of protocols.

Based on that lack the taxonomy has been slightlymodified by adding such features as **type of cast** and**cost function**. Type of cast feature is an upper levelclassification and so the protocols to be classified must firstly divide by type of cast and after that the moreaccurate taxonomy can be applied. The above mentionedtaxonomy is applied to unicast protocols, while in the context of multicast and geocast protocols a specified taxonomy has been introduced. The overall taxonomyand specially the unicast protocol classification can beseen in figure 1.

The cost function is a classification to be concatenated after presented taxonomy. It is like a remark to benoticed when considering the applicability of theprotocol to be chosen.

A. *Communication Model*

Protocols can be divided according to communications model to protocols that are designed for **multi-channel**or **single-channel** communications. Multi-channelprotocols are routing protocols generally used in TDMA or CDMA-based networks. They combine channelassignment and routing functionality. That kind ofprotocol is e.g. Cluster head Gateway Switched Routing (CGSR) [4].
Single -channel protocols presume one shared media to be used. They are generally CSMA/CA-oriented, butthey have a wide diversity in which extend they rely onspecific link-layer behaviors.

B. *Structure*

Structure of a network can be classified according to node uniformity. Some protocols treat all the nodesuniformly, other make distinctions between differentnodes. In **uniform protocols** there is no hierarchy innetwork, all nodes send and respond to routing controlmessages at the same manner.

In **non-uniform protocols** there is an effort to reduce the control traffic burden by separating nodes in dealingwith routing information. Non-uniform protocols fallinto two categories: protocols in which each nodefocuses routing activity on a subset of its neighbors andprotocols in which the network is topologically partitioned. These two different methods for non-uniformityare called **neighbor selection** and**partitioning** respectively.

With neighbor selection mechanism, every node has its own criteria to classify network nodes to near or toremote nodes.

In partitioning protocols thatdifferentiation is to use hierarchical node separation.Hierarchical protocols have some upper-level and lower level nodes and certain information difference between them.

C. *State Information*

Protocols may be described in terms of the state information obtained at each node and / or exchangedamong nodes. **Topology-based protocols** use theprinciple that every node in a network maintains largescaletopology information. This principle is just the same as link-state protocols use.
Destination-based protocols do not maintain large-scale topology information. They only may maintain topologyinformation needed to know the nearest neighbors. The best known such protocols are distance-vector protocols,which maintain a distance and a vector to a destination(hop count or other metric and next hop).

D. *Scheduling*

The way to obtain route information can be a continuous or a regular procedure or it can be trigged only by on demand. On that basis the protocols can be classified toproactive and on-demand protocols. **Proactiveprotocols**, which are also known as table-driven protocols, maintain all the time routing information for all known destinations at every source. In these protocolsnodes exchange route information periodically and / or in response to topology change.

In on-demand i.e. in **reactive protocols** the route is only calculated on demand basis. That means that there is nounnecessary routing information maintained. The routecalculation process is divided to a route discovery and a route maintenance phase. The route discovery process isinitiated when a source needs a route to a destination.
The route maintenance process deletes failed routes andre-initiates route discovery in the case of topologychange.

E. *Type of Cast*

Protocols can be assumed to operate at unicast, multicast, geocast or broadcast situations.

In **unicast protocols** one source transmits messages or data packets to one destination. That is the most normaloperation in any network. The unicast protocols are alsothe most common in ad hoc environment to bedeveloped and they are the basis on which it is a possibility to construct other type of protocols. Unicast protocols have thought some lacks when there is a needto send same message or stream of data to multipledestinations. So there is an evitable need for multicast protocols.

Multicast routing protocols try to construct a desirable routing tree or a mesh from one source to severaldestinations. These protocols have also to keep up withinformation of joins and leave ups to a multicast group.

The purpose of **geocast protocols** are to deliver data packets for a group of nodes which are situated on at specified geographical area. That kind of protocol can also help to alleviate the routing procedure by providinglocation information for route acquisition.

Broadcast is a basic mode of operation in wireless medium. Broadcast utility is implemented in protocols asa supported feature. Protocol only to implementbroadcast function is not a sensible solution. That is the reason not to classify protocols to broadcast protocols.But it is worth to mention if a protocol is not supporting that method.

F. *Cost Function*

When making routing decisions in ad hoc environments, it is normally not enough to take only considerations tohop count. In ad hoc networks there is a wide variety ofissues to consider such as link capacity, which can vary in large scale, latency, link utilization percentage and terminal energy issues to mention a few most relevant.That is why there is a need to adapt cost functions toroute calculations.

Rough classification of protocols according to cost function can be based on **hop count** approach (no special cost function applied) and to **bandwidth** or**energy** based cost functions. Also quite a differentapproach to routing metrics is used by AssociativityBased Routing (ABR) protocol, which uses **degree of association stability** for a metric to decide for a route.That means that presumably more permanent routes arepreferred.

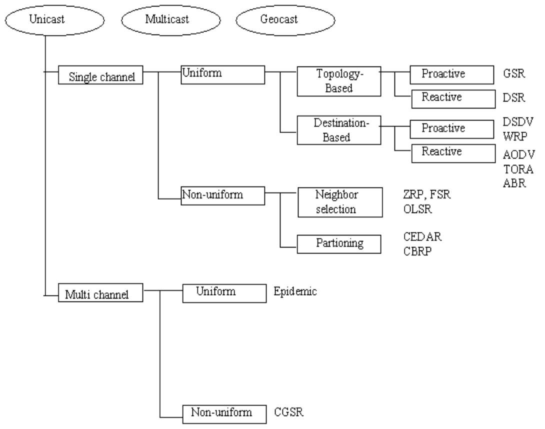

Figure1: Taxonomy of Protocols. Classification ofunicast protocols shown.

III. OVERVIEW OF SELECTED PROTOCOLS

There are unicast, single channel protocols, which are uniform or non-uniform. Uniform protocols are dividedto topology-based protocols, in where nodes are aware ofthe topology information of all other nodes in thenetwork or to destination-based protocols, in where nodes only know the preferred next hop to a destination.One protocol to belong to that topology-based class isGSR (Global State Routing) and the other is DSR (Destination Source Routing). One main differencebetween these protocols is the scheduling method. GSR is a proactive protocol, which will all the time have theinformation needed for routing. DSR is on its behalf areactive protocol, which will obtain needed information only on demand.

To destination-based protocols belong such protocols as DSDV, AODV, TORA, ABR and WRP. The wellknowndifference between e.g. DSDV and AODV is thescheduling method. The DSDV is proactive as is WRP, but AODV, TORA and ABR all are reactive protocols.

To be classified to single channel, non-uniform protocols there are such protocols as ZRP, FSR, OLSR, CEDARand CBRP. Form these protocols ZRP, FSR, and OLSR belong to neighbor selection protocols, which have a common feature to select network subsets by individualnodes themselves. In partitioning protocols there aresome kind of clustering and cluster head selectionmechanism. To partitioning protocols belongs e.g. CEDAR and CBRP.

To unicast multi-channel protocols include such protocols as CGSR and Epidemic. CGSR is a non-uniformprotocol and Epidemic is a uniform protocol.

The unicast protocols presented here shortly are the following:

- GSR
- WRP
- OLSR
- FSR
- CEDAR
- CGSR
- Epidemic

A. *Light-Weight Proactive Source Routing Protocol*

1) *Design of PSR:*

PSR provides every node with a breadth-first spanning tree (BFST) of the entire network rooted at itself. To do that, nodes periodically broadcast the tree structure to their best knowledge in each iteration. Based on the information collected from neighbors during the most recent iteration, a node can expand and refresh its knowledge about the network topology by constructing a deeper and more recent BFST. This knowledge will be distributed to its neighbors in the next round of operation. On the other hand, when a neighbor is deemed lost, a procedure is triggered to remove its relevant information from the topology repository maintained by the detecting node. Intuitively PSR has about the same communication overhead as distance vector- based protocols. We go an extra mile to reduce the communication overhead incurred by PSR’s routing agents. Details about such overhead reduction will be discussed in following section.

Before describing the details of PSR, we will first review some graph theoretic terms used here. Let us model the network as an undirected graph $G = (V;E)$, where V is the set of nodes (or vertices) in the network and E is the set of wireless links (or edges). Two nodes u and v are connected by an edge $e = (u; v) > E$ if they are close to each other and can communicate directly with a given reliability. Given a node v, we use $N(v)$to denote its open neighborhood, *i.e.*, $\{u > V\ |(u; v) > E\}$. Similarly, we use $N[v]$ to denote its closed neighborhood, *i.e.*, $N(v)\cup\{v\}$. The readers are referred to the monograph of West [14] for other graph theoretic notions.

2) *Route update:*

Due to its proactive nature, the update operation of PSR is iterative and distributed among all nodes in the network. At the beginning, a node v is only aware of the existence of itself, so there is only a single node in its BFST, which is the root node v. By exchanging the BFSTs with the neighbors, it is able to construct a BFST within $N[v]$, *i.e.*, the star graph centered at v, denoted by S_v.

In each subsequent iteration, nodes exchange their spanning trees with their neighbors. From the perspective of node v, towards the end of each operation interval, it has received a set of routing messages from its neighbors packaging the BFST's Node v incorporates the most recent information from each neighbor to update its own BFST. It then broadcasts this tree to its neighbors at the end of the period. Formally, v has received the BFSTs from some of its neighbors. Including those from whom v has received updates in recent previous iterations, node v has a BFST, denoted Tu, cached for each neighbor $u > N(v)$. Node v constructs a union graph

$$G_v = S_v \cup \bigcup_{u \in N(v)} (T_u - v)$$

Here, we use $T - x$ to denote the operation of removing the sub-tree of Trooted at node x. As special cases, $T - x = T$ if x is not in T and $T - x = \emptyset$ if x is the root of T. Then, node v calculates a BFST of G_v, denoted T_v, and places $T_v$in a routing packet to broadcast to its neighbors.

The above update of the BFST happens multiple times within a single update interv also that a node can incorporate new route information to its knowledge base more quickly. To the extreme, $Tv$is modified every time a new tree is received from a neighbor. Apparently, there is a trade-off between the routing agent's adaptivity to network changes and computational cost. Here, we choose routing adaptivity as a higher priority assuming that the nodes are becoming increasingly powerful in packet processing. Nevertheless, this does not increase the communication overhead at all because one routing message is always sent per update interval.

Assume that the network diameter, *i.e.*, the maximum pairwise distance, is D hops. After D iterations of operation, each node in the network has constructed a BFST of the entire network rooted at itself, since nodes are timer-driven and thus synchronized. This information can be used for any source routing protocol. The amount of information that each node broadcasts in an iteration is bounded by $O(|V|)$and the algorithm converges in D iterations.

3) *Neighborhood trimming:*

The periodically broadcast routing messages in PSR also double as "Hello" messages for a node to identify which other nodes are its neighbors. When a neighbor is deemed lost, its contribution to the network connectivity should be removed, called "neighbor trimming". Consider node v. The neighbor trimming procedure is triggered at v about neighbor u when

- No routing update or data packet has been received from this neighbor for a given period of time, or
- A data transmission to node u has failed as reported by the link layer.Node$v$ responds by
- first updating $N(v)$ with $N(v)-\{u\}$,
- next constructing the union graph with the information of u removed.

$$G_v = S_v \cup \bigcup_{w \in N(v)} (T_w - v)$$

- and then computing the BFST Tv.

Notice that Tvthus calculated is not broadcast immediately to avoid excessive messaging. With this updated BFST at v, it is able to avoid sending data packets via lost neighbors. Thus, multiple such neighbor trimming procedures may be triggered within one period.

4) *Streamlined differential update:*

In addition to dubbing route updates as hello messages in PSR, we interleave the "full dump" routing messages as stated previously with "differential updates". The basic idea is to send the full update messages less frequently than shorter messages containing the difference between the current and previous knowledge of a node's routing module. Both the benefit of such an approach and how to balance between these two types of messages have been studied extensively in earlier proactive routing protocols. In this work, we further streamline the routing update in two new avenues. First, we use a compact tree representation in full-dump and differential update messages to halve the size of these messages. Second, every node attempts to maintain an updated BFST as the network changes so that the differential update messages are even shorter.

Compact tree representation: For the full dump messages, our goal is to broadcast the BFST information stored at a node to its neighbors in a short packet. To do that, we first convert the general rooted tree into a binary tree of the same size,say $s$nodes, using leftchild sibling representation. Then we serialize the binary tree using a bit sequence of $34 \times s$ bits, assuming IPv4 is used. Specifically, we scan the binary tree layer by layer. When processing a node, we first include its IP address in the sequence. In addition, we append two more bits to indicate if it has the left and/or right child.
For example, the binary tree in Figure 2 is represented as A10B11C11D10E00F00G11H00I00. As such, the size of the update message is a bit over half compared to the

traditional approach, where the message contains a discrete set of edges.

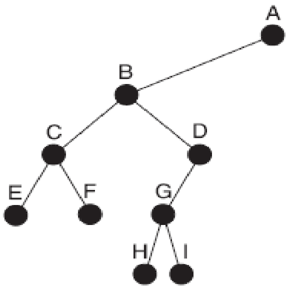

Fig.2 Binary Tree

The difference between two BFSTs can be represented by the set of nodes who have changed parents, which are essentially a set of edges connecting to the new parents. We observe that these edges are often clustered in groups. That is, many of them form a sizeable tree subgraph of the network. Similar to the case of full dump, rather than using a set of loose edges, we use a tree to package the edges connected to each other. As a result, a differential update message usually contains a few small trees, and its size is noticeably shorter.

Stable BFST — The size of a differential update is determined by how many edges it includes. Since there can be a large number of BFSTs rooted at a given node of the same graph, we need to alter the BFST maintained by a node as little as possible when changes are detected. To do that, we modify the computation described earlier in this section such that a small portion of the tree needs to change either when a neighbor is lost or when it reports a new tree.

Consider node *v* and its BFST T_u. When it receives an update from neighbor *u*, denoted by T_u, it first removes the subtree of $T_v$rooted at *u*. Then it incorporates the edges of *Tu*for a new BFST. Note that the BFST of $(T_v{-}u) \cup T_v$may not contain all necessary edges for *v* to reach every other node. Therefore, we still need to construct the union graph

$$(T_v - u) \cup \bigcup_{w \in N(v)} (T_w - v)$$

Before calculating its BFST. To minimize the alteration to the tree, we add one edge of $T_u{-}v$ to $T_v{-}u$ at a time. When node *v* thinks that a neighbor *u* is lost, it deletes the edge (*u; v*) but still utilizes the network structure information contributed by *u* earlier. That is, even if it has moved away from *v*, node *u* may still be within the range of one of *v*'s neighbors. As such, $T_v$should be updated to a BFST of

$$(T_v - u) \cup (T_u - v) \cup \bigcup_{w \in N(v)} (T_w - v) .$$

B. *Estimated distance-based routing protocol*

In this section, first, we introduce a method to amend the EGD. The amended EGD using the ETD is the EstD, whichis used as a metric to decide whether a node is a good next-hop candidate. In the third part of this section, we propose the EDRP.

1) *Evaluating the LQ Using EGD:*

The calculation method of the EGD naturally has the ability to evaluate the LQ. Each node knows the EGDs of its neighbor nodes, and thus, from the first two relations of (1), the node can solve the relative velocity *v* with its neighbor nodes. Then, we obtain $v = \sqrt{A}$, where *A* is shown in (2).
We use *D*2 and *D*1 to determine whether a neighbor node is coming nearer or going away. If the neighbor is coming nearer, then this link may be strong; otherwise, we use a simple method to estimate the LQ. While receiving a packet from a neighbor node, the relative velocity *v* and the distancesD_2 andD_1 can be calculated in the EGD calculation part. In addition, the distance can be calculated by the RSS when receiving the packet. Then, the algorithm is shown in Algorithm .1.

Algorithm 1 LinkQuality(*pkt*)

```
if $D_2 \le D_1$ then
    return Strong;
else
    Dis = getDistanceByRSS(pkt);
    Q = (Tx_Radius− Dis)/v;
    if LQ <STRONG_LINK_THRESH then
        return Weak;
    else
        return Strong;
    end if
end if
```

In function LinkQuality(*pkt*), Tx_Radius is the transmission radius of a node, and STRONG_LINK_THRESH is a threshold that is used to determine whether the link is strong or weak. This function can be used by the node to evaluate the LQ with its neighbor node when receiving a *pkt*. This LQ is used to determine whether this node is a valuable candidate to forward an RREQ packet, because selecting a weak link may lead to a path disconnection a short time later.

2) *Amending the EGD Using ETD:*

Using the EGD to estimate the actual distance is effective when it is less than the E-Radius. As time goes by, if the two nodes do not encounter for the second time, then the EGD may become very large, which cannot correctly estimate the actual distance. For a large EGD, we need a criterion to check its effectiveness. Using RSS, we can estimate the distance between neighbors, and in multi-hop networks, the sum of the distance of every hop can be used as a criterion. This distance is called the ETD because it is on the topology of the network and is not a geometrical distance. The ETD is computed as follows

- We modify the RREQ and route reply (RREP) headers and add a new field ETD. When a source node sends an RREQ packet, it sets to ETD = 0.

The intermediate node receives this packet and can calculate the distance d from the previous hop node and then sets to ETD = ETD + d.

- To reduce the error of EGD and ETD, if EGD <ETD, then we determine that the EGD is effective, and then we use EGD to substitute the ETD. On the other hand, if ETD <EGD, we determine that the EGD is in-effective, and then we set to EGD = ∞.
- The destination node can estimate the topological distance to the source node. When the destination node replies an RREP packet to the source node, it does a similar procedure as the source node, and so do the intermediate nodes and the source node when receiving an RREP packet.
- The source node can also estimate the topological distance to the destination node. As historical information, the ETD has a lifetime ETD_LIFETIME. Thus, the ETD is also a function of time. If the ETD_LIFETIME expires, then we set ETD = ∞.
- From the computation of the ETD, we know that if EGD<ETD, then the EGD is effective, and if ETD <EGD, then ETD is effective. Therefore, the EstD should be the minimum of EGD and ETD, which is defined as follows:

$$EstD = \min\{EGD, ETD\}.$$

3) *Protocol Description:*

In the route discovery, when an intermediate node receives an RREQ packet, first, if the node has seen this packet, then it directly discards this packet. Then the node determines the LQ for the previous hop node; if the link is weak, the node also discards the packet. After doing that, the node determines its forwarding strategy for the RREQ packet according to its own EstD metric, and it is divided into three situations (we use EstD(*u*) and ETD(*u*) to denote EstD and ETD for node *u*, respectively):

1. If a node is in the dst-Zone, then the EstD(*dst*) used to estimate the actual distance is credible. Therefore, this node can execute greedy forwarding according to EstD(*dst*) to steer the RREQ packet toward the destination. When the EstD(*dst*) of the current node is less than the EstD(*dst*) carried in the RREQ packet, the RREQ packet is forwarded; otherwise, it is discarded.

2. If a node is not in the dst-Zone but is in the src-Zone, the node cannot use the EstD(*dst*) because it may be incredible. Thus, it uses ETD(*dst*) to determine whether it has some historical information. If ETD(*dst*) is less than ∞, that means in recent time, this node has a route to the destination Therefore, when the ETD(*dst*) of the source node is less than∞, based on the spatial locality, if the intermediate node whose ETD(*dst*) is also less than ∞, then it may be in the same side with the destination node, relative to the source node. Thus, this node should use a high probability to forward the RREQ packet. Otherwise, if the intermediate node whose ETD(*dst*) is ∞, then it may be in the reverse side with the destination

3. Except for the preceding two situations, only when the node in the mid-Zone between src-Zone and dst-Zone, it needs to forward the RREQ packet. For the source node, if EstD(*dst*) ≤ E-Radius, then we use IN_1_ZONE to denote the position of the source node; if E-Radius <EstD(*dst*) ≤ 2E-Radius, then we use IN_2_ZONE todenote the position of the source node; if EstD(*dst*) >2E-Radius, then that means there exists mid-Zone, and we use IN_3_ZONE to denote the position of the source node

Algorithm.2 Forward(*pkt*)

Definitions:
Gossip(*pkt*): A gossip algorithm for optimization of broadcasting
Random(0, 1): Return a random number between [0, 1)
***P*max, *P*min**: The high and low probabilities used to forward the RREQ packet
***p*thresh**: The probability calculated for forwarding the RREQ packet

```
if pkt.id already seen then
    discard(pkt); return;
end if

if LinkQuality(pkt) == Weak then
    discard(pkt); return;
end if

if EstD(dst) ≤ E-Radius then
    {The RREQ has entered the dst-Zone, and in this zone the estimation for the dst is    credible.}
    if EstD(dst) <pkt.EstDthen
        pkt.EstD = EstD(dst);
        broadcast(pkt); return;
    else
        discard(pkt); return;
    end if
else if EstD(src) ≤ E-Radius
    {The RREQ is still in the src-Zone.}
    if pkt.src.ETD< ∞ then
        if ETD(dst) < ∞ then
            pthresh = Pmax;
        else
            pthresh = Pmin;
        end if
    else
        pthresh = Gossip(pkt);
    end if
else
```

```
{The RREQ has left the src-Zone, but it does not enter the dst-Zone.}
    if pkt.src.Pos == IN_3_ZONE then
        pthresh = Gossip(pkt);
    else
        discard(pkt); return;
    end if
end if

if Random(0, 1) <pthreshthen
    pkt.EstD = EstD(dst);
    broadcast(pkt); return;
else
    discard(pkt); return;
end if
```

IV. CANCLUSION

The presented taxonomy of routing protocols is a meaningful attempt to clarify the vast field of ad hoc routing protocols. It is so because it tries to reveal the main design and implementation principles behind protocols. The taxonomy is a little bit complicated and it

is not always an easy task to classify a protocol according to that taxonomy, but the meaning of classifying is try to get some rough basis for protocol's performance evaluation. It should be assumed that same kinds of protocols behave quite the same way in simulations.

When choosing a protocol to a specified network one should consider the following issues:

- What is the size of the network. If the network could be considered or forecasted to be large the chosen protocol should support scaling issues.
- What is the degree of mobility; how often links are assumed to cut off. Some protocols (usually reactive) have better performance over some other protocols (usually proactive) when mobility is high
- What are the requirements of user applications for the underlying network. Real-time applications require quite different services compared to non-time critical message delivery.

When the network structure and the node behaviors are understood, the right or at least near optimal protocol could be chosen. It is quite inevitable that inside the same network many different protocols should be implemented to cover all the networks states. Some kind of mixture of mutually compatible protocols could be needed. The other way to reach the goal is that protocols will merge and form a protocol, which has all the wished properties, but none of the weak ones. This can be a way to make a giant protocol to be good at theory, but in practice not a viable solution.

REFERENCES

[1] E.M. Royer, C-K. Toh, A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks, IEEE Personal Communications Magazine, April 1999, pp. 46-55. http://www.cs.ucsb.edu/~eroyer/publications.html

[2] N. Nikaein, H. Labiod, C. Bonnet, "DDRDistributed Dynamic Routing Algorithm for Mobile Ad-Hoc Networks", Proceedings of the First Annual Workshop on Mobile Ad Hoc Network&Computing, MobiHOC 2000, Boston,pages 19-27, August 2000. http://www.eurecom.fr/~nikaeinn/ddr.pdf

[3] L.M. Feeney: "A Taxonomy for Routing Protocols in Mobile Ad Hoc Networks", SICS Technical Report T99/07, October 1999. http://www.sics.se/~lmfeeney/research.html

[4] I. Chlamtac, M. Conti, and J.-N. Liu, "Mobile Ad hoc Networking: Imperatives and Challenges," *Ad Hoc Networks*, vol. 1, no. 1, pp. 13–64, July 2003.

[5] M. Al-Rabayah and R. Malaney, "A New Scalable Hybrid Routing Protocol for VANETs," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 6, pp. 2625–2635, July 2012.

[7] R. Rajaraman, "Topology control and routing in ad hoc networks: A survey," *Special Interest Group on Algorithms and Computation Theory (SIGACT) News*, vol. 33, pp. 60–73, June 2002.

[8] Y. P. Chen, J. Zhang, and I. Marsic, "Link-Layer-and-Above Diversity in Multi-Hop Wireless Networks," *IEEE Communications Magazine*, vol. 47, no. 2, pp. 118–124, February

[9] Internet Society C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc ondemand distance vector (AODV) routing," RFC 3561, Internet Society, Jul. 2003.

[10] IETF Trust D. Johnson, Y. Hu, and D. Maltz, "The dynamic source routing protocol for mobile ad hoc networks (DSR) for IPv4," RFC 4728, IETF Trust, Feb. 2007.

[11] Z. Haas, J. Y. Halpern, and L. Li, "Gossip-based ad hoc routing," in *Proc. IEEE INFOCOM*, 2002, vol. 21, pp. 1707–1716.

[12] X. Li, K. Moaveninejad, and O. Frieder, "Regional gossip routing for wireless ad hoc networks," *Mobile Netw. Appl.*, vol. 10, no. 1/2, pp. 61– 77, Feb. 2005.