CHAPTER 2

REVIEW OF LITERATURE

2.1 INTRODUCTION

Wireless communication technology has been developed with two primary models. One is fixed infrastructure based model in which much of the nodes are mobile and connected through fixed backbone nodes using wireless medium. Another model is Mobile Ad-hoc network. Mobile Ad-Hoc Networks (MANETs) are comprised of Mobile Nodes (MNs) that are self-organizing and cooperative to ensure efficient and accurate packet routing between nodes (and, potentially, base stations). There are no specific routers, servers, access points for MANETs. Because of its fast and easy deployment, robustness, and low cost, typical MANET applications could be found in the following areas like Military applications (i.e. a temporary network in the battlefield), Search and rescue operations, Temporary networks within meeting rooms, airports, Vehicle-to-vehicle communication in smart transportation, Personal Area Networks connecting mobile devices like “mobile phones”, laptops, smart watches, and other wearable computers etc. Design issue for developing a routing protocol for wireless environment with mobility is very different and more complex than those for wired network with static nodes (Murthy 1996). Main problems in mobile ad hoc network are Limited bandwidth and frequent change in the topology. Although there are lots of routing protocols that can be used for unicast and multicast communication within the Mobile Ad hoc networks, it is observed that any one protocol cannot fit in all the different scenarios,
different topologies and traffic patterns of Mobile Ad-Hoc Networks applications. For instance, proactive routing protocols are very useful for a small scale MANETs with high mobility, while reactive routing protocols are very useful for a large-scale MANETs with moderate or less topology changes. Hybrid routing protocol attempts to strike balance between the two such as proactive for neighborhood and reactive for far away (Ismail 2001).

Multicast in another category of routing protocol in MANETS despite this, which supports with great efficiency, the group communication with the high throughput. The benefits of the use of multicasting within MANETS are of many told. It can decrease the cost of wireless communication and increase the efficiency and throughput of the wireless link between two nodes whenever we are sending multiple copies of the same messages by accomplishing the inherent broadcasting properties of wireless transmission. In place of sending same data through multiple unicasts, multicasting decreases channel capacity consumption, sender nodes and routers processing, energy utilization, and data delivery delay, which are deliberate important for MANETS. If the mobile nodes in the MANET move too quickly, they have to repair in order to broadcast to achieve node to node communication. Every routing protocol has its advantages and disadvantages, and aims at a specific application. Finally, the expected standard for routing protocols in the Mobile Ad-Hoc Networks is very likely to combine some of the most competitor schemes. Thus the goal for a routing protocol is to minimize its control traffic overhead while at the same time; it should be capable of rapidly linking failure and addition caused by node movements (Haas 1998).
2.2 MANETs

This section provides an overview of MANETs about their characteristics, advantages, challenges and applications.

2.2.1 MANET Characteristics

Following are few of the characteristics of MANETs (Aarti 2013)

**Distributed Operation:** There is no Background Network with regard to the central control of the network Operation. The network control gets distributed among the nodes. The nodes involved in a MANETS are expected to cooperate with one another and communicate among themselves. Every node will act as relay as per the need to implement certain functions such as routing and security.

**Multi hop routing:** The packet should be forwarded via one or more intermediate nodes when a node tries to send information to other nodes which are out of communication range.

**Autonomous Terminal:** Each mobile node is an independent one in MANETS which could both act as a host and a router as well.

**Dynamic Topology:** Nodes freely move with different speeds resulting in the random change of the Network Topology. The time can’t be predicted. The nodes in the MANETS establish their own.

**Light–Weight Terminals:** In maximum cause, the nodes in the MANETS are mobile with inefficient in terms of CPU capability, Storage and Memory.
Shared Physical Medium: The wireless communication medium is open for access to any entity with the appropriate equipment and adequate resources the Access to the channel cannot to restrict accordingly.

2.2.2 MANET Advantages

The advantages of an Ad-Hoc network include the following: (Tyagi 2013)

- They provide access to information and services regardless of geographic position.
- Independence from central network administration. Self-configuring network, nodes act as routers. It is less expensive as compared to wired network.
- Scalable—accommodates the addition of more nodes
- Improved Flexibility
- Robust due to decentralize administration
- The network can be set up at any place and time

2.2.3 MANET Challenges

MANET Challenges could be summarized as below: (Aarti 2013)

Limited bandwidth: Wireless link continues to have significantly lower capacity than networks with infrastructure. In addition, the realized throughput of wireless communication after accounting for the effect of multiple access, fading, noise, and interference conditions, etc., is often much less than a radio’s maximum transmission rate.
Dynamic topology: Dynamic topology membership may disturb the trust relationship among nodes. The trust may be disturbed if some nodes are detected as compromised.

Routing Overhead: In wireless ad hoc networks, nodes often change their location within network. So, some stale routes are generated in the routing table which leads to unnecessary routing overhead.

Hidden terminal problem: The hidden terminal problem refers to the collision of packets at a receiving node due to the simultaneous transmission of those nodes that are not within the direct transmission range of the sender, but are within the transmission range of the receiver.

Packet losses due to transmission errors: Ad hoc wireless networks experiences a much higher packet loss due to factors such as increased collisions due to the presence of hidden terminals, presence of interference, uni-directional links and frequent path breaks due to mobility of nodes.

Mobility-induced route changes: The network topology in an ad hoc wireless network is highly dynamic due to the movement of nodes; hence an on-going session suffers frequent path breaks. This situation often leads to frequent route changes.

Battery constraints: Devices used in these networks have restrictions on the power source in order to maintain portability, size and weight of the device.

Security threats: The wireless mobile ad hoc nature of MANETs brings new security challenges to the network design. As the wireless medium is
vulnerable to eaves dropping and ad hoc network functionality is established through node cooperation, mobile ad hoc networks are intrinsically exposed to numerous security attacks.

2.2.4 MANET Applications

Some of the typical applications include: (Tyagi 2013)

**Military battlefield:** Ad-Hoc networking would allow the military to take advantage of common place network technology to maintain an information network between the soldiers, vehicles, and military information head quarter.

**Collaborative work:** For some business environments, the need for collaborative computing might be more important outside office environments than inside and where people do need to have outside meetings to cooperate and exchange information on a given project.

**Local level:** Ad-Hoc networks can autonomously link an instant and temporary multimedia network using notebook computers to spread and share information among participants at a conference or classroom. Another appropriate local level application might be in home networks where devices can communicate directly to exchange information.

**Personal area network and Bluetooth:** A personal area network is a short range, localized network where nodes are usually associated with a given person. Short-range MANET such as Bluetooth can simplify the inter communication between various mobile devices such as a laptop, and a mobile phone.
Commercial Sector: Ad hoc can be used in emergency/rescue operations for disaster relief efforts, e.g. fire accidents and natural calamities like, flood, or earthquake. Emergency rescue operations must take place where non-existing or damaged communications infrastructure and rapid deployment of a communication network is needed.

2.3 ROUTING METRICS

Routing is the most vital functionality to direct communication in large networks. Desirable paths are determined based on routing metrics (Baumann 2007). These are then used to compare routing protocols. This section provides a high level taxonomy for categorizing and classifying routing metrics. Routing metrics could be broadly classified into qualitative and quantitative metrics. Packet Delivery Ratio, Average Delay, Throughput are few quantitative metrics. For a good routing protocol, throughput and packet delivery ratio should be high whereas average delay should be less.

The common design approach for MANET routing protocols has been to apply the same routing paradigms conceived for traditional wired networks. This design choice implicitly assumes that wireless links are similar to wired links, and that they can be represented as point-to-point connections. However, mobility and link fluctuations create topology changes that the routing protocol needs to detect and handle in order to maintain operational routes needed by the applications. If the routing protocol is too slow in acting on topology changes, the result is inevitably problems for the applications, due to for instance packet loss, delay or jitter.
<table>
<thead>
<tr>
<th>Function</th>
<th>Routing Metric</th>
<th>Optimization Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traffic Based</td>
<td>Delay/Latency</td>
<td>Minimize average end-to-end delay</td>
</tr>
<tr>
<td></td>
<td>Delay Variation/Jitter</td>
<td>Minimize delay of streams</td>
</tr>
<tr>
<td></td>
<td>Queue length</td>
<td>Minimize delay, load-balancing</td>
</tr>
<tr>
<td></td>
<td>Bandwidth/throughput</td>
<td>Maximize Throughput, load-balancing</td>
</tr>
<tr>
<td></td>
<td>Packet Loss Ratio (PLR)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Delivery rate</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Expected Transmission Time (ETT)</td>
<td>Maximize throughput</td>
</tr>
<tr>
<td></td>
<td>Expected Transmission Count (ETX)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fruitful Number of Communications (FNC)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Expected Data Rate (EDR)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Packet Reordering Ratio (PRR)</td>
<td>Minimize delay</td>
</tr>
<tr>
<td>Radio Information</td>
<td>Signal Strength</td>
<td>Minimize transmission energy / Minimize interference</td>
</tr>
<tr>
<td></td>
<td>Signal-to-Noise Ratio (SNR)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Estimated Transmission Time (ETT)</td>
<td>Minimize the time, a packet occupies the</td>
</tr>
<tr>
<td></td>
<td>Medium Time Metric (MTM)</td>
<td>Medium</td>
</tr>
</tbody>
</table>
Table 2.1 (Continued)

<table>
<thead>
<tr>
<th>Function</th>
<th>Routing Metric</th>
<th>Optimization Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Topology</td>
<td>No. of neighbours</td>
<td>Minimize number of hops</td>
</tr>
<tr>
<td></td>
<td>No. of Paths to a Node</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Hop count</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Minimal Hop Count</td>
<td></td>
</tr>
<tr>
<td>Mobility &amp;</td>
<td>Geographical distance</td>
<td>Find “short” path</td>
</tr>
<tr>
<td>Geography</td>
<td>Speed of nodes</td>
<td>Find stable paths</td>
</tr>
<tr>
<td></td>
<td>Link lifetime</td>
<td>Maximize long-term accuracy of a path, minimize recalculations of path</td>
</tr>
<tr>
<td>Energy Based</td>
<td>Energy consumed by each payload</td>
<td>Minimize the total power consumption</td>
</tr>
<tr>
<td></td>
<td>Communication energy</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Minimal Total Power metric (MTPR)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Remaining battery capacity</td>
<td>Maximize the network lifetime</td>
</tr>
<tr>
<td></td>
<td>Packet Loss Ratio</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Interference</td>
<td></td>
</tr>
</tbody>
</table>

Table 2.1 lists the various MANET routing metrics along with their optimization goal. This classification is based on their functional usage. The main aim of this taxonomy is to bring structure into the complex area of routing metrics. It helps in decision making with regards to the choice of suitable metrics in a given problem.

2.4 CLASSIFICATIONS OF MANET ROUTING PROTOCOLS

The primary task of the routing protocol is to discover and maintain routes to needed network destinations. A great number of routing protocols for
MANETs have been proposed, dealing with these challenges in very different ways, and the classification of such a diverse group of protocols is equally varying. Many of the challenges in MANETs must be dealt with by the routing protocol:

- Limited capacity, demanding low routing protocol overhead
- Varying link quality with regards to bit error rate
- Mobility, leading to link breaks and new links
- Distributed routing, where it cannot be guaranteed that all network nodes have the same view of the topology

As far as the ad-hoc network is concerned there are two types of routing protocols. They are the table driven and demand routing protocols. Table driven protocols are otherwise known as proactive routing protocol. With regard to table driven protocol each node maintains in advance. Routing information gets updated periodically and exchanged among the neighbours a large number of overhead bits will be generated during periodic exchange of routing table information.

On the other hand reactive routing protocol determines the routes as and when it is required by a node in the network. On demand route creation significantly reduces the control overhead. Routes are determined by sending route request packet to the immediate neighbours in response to the route request packet intermediate nodes or the destination replied by uncasting the route reply packet. AODV & DSR are the on demand routing protocols. The aim of any routing protocol, regardless of its classification, is to provide efficient routing with a minimum of overhead. This high level of categorization is shown in Figure 2.1 (Kamal Kant)
Another classification of multicast protocols would be:

- **Stateless multicast** – Ex: DDM, E2M
- **Tree-based multicast** – Ex: MAODV, MOLSR
- **Mesh-based multicast** – Ex: CAMP, ODMRP
- **Hybrid multicast** – Ex: MANSI, AMRoute and
- **Flooding**

Stateless multicast is a form of multicast where the source alone keeps track of which nodes are subscribers to the information. This type of multicast protocol is suited for small multicast groups. Stateless multicast is suitable as long as the number of receivers is small, but with a higher number of receivers, the destination overhead in each packet may be excessive.

In such scenarios, tree-based multicast protocols may be more suitable. The tree-based protocols are best suited for static networks, mainly because the trees
offer no redundant distribution paths, and in the event that a link is broken, the packet flow to the downstream nodes is interrupted. The tree-based protocols are further classified into source-based-tree and shared tree protocols. The source-based tree protocols create and maintain separate trees spanning from each multicast source, while shared tree protocols maintain one multicast tree for all sources.

Considering the problems that occur with tree-based multicast protocols, mesh-based protocols have been proposed as an alternative. Mesh-based protocols provide redundant links in the forwarding paths, increasing the probability that packets reach all destinations, even if a link break is occurring. The cost of redundancy is the increased overhead, due to both signalling the increased number of links in the forwarding paths, and the redundant data transmissions. Whereas the tree and mesh based protocols have different ways to solve multicast forwarding using single or multiple paths, the hybrid protocols seek to pair the structure and efficiency of the tree-based protocols with the robustness of the mesh protocols.

Figure 2.2 provides the classification of unicast routing protocols. In routing protocols based on flat routing, all nodes that participate in routing play an equal role. The hierarchical protocol class is the complementary class of the flat routing, addressing scalability. In this class, nodes play different roles in the routing, where nodes are grouped and routing information is restricted on the basis of these groups. Protocols of the third class, geographical routing, require all nodes to know their exact position, for example using the Global Positioning System (GPS). Based on this knowledge, routing requests may, for example, be restricted to the geographical area where the shortest path to the destination is expected to exist.
2.4.1 Unicast Routing Protocols

Most applications in the MANET are based upon unicast communication. Thus, the most basic operation in the IP layer of the MANET is to successfully transmit data packets from one source to one destination. The forwarding procedure is very simple in itself: with the routing table, the relay node uses the destination address in the data packet to look it up in the routing table. If the longest matching destination address is found in the table, the packet is sent to the corresponding next hop. The problem that arises is how the routing table is built in the nodes in the MANET (Murthy 1996). Figure 2.3 shows the unicast process. In the unicast routing one separate copy of data is sent to each receiver from the source node. Data packet is replicated at the sender node and then delivered to each destination node. By this process we can easily see that bandwidth is consumed by the redundant data packets. Many applications use the unicast routing protocol depending upon the need of the
application. There are proactive, reactive and hybrid routing protocol in unicast routing for Mobile distributed networks.

![Unicast Process Diagram]

**Figure 2.3 Unicast Process**

### 2.4.1.1 Proactive unicast routing protocols

Traditional routing protocols such as Optimized Link State Routing Protocol (OLSR), The Fisheye State Routing (FSR), and Topology Broadcast Based on Reverse-Path Forwarding Routing Protocol (TBRPF) are proactive unicast routing protocols. Periodic broadcast of network topology updates (e.g., distance vector or link state information) is necessary to compute the shortest path from the source to every destination, which consumes a lot of bandwidth. Although they are widely used in the Internet backbone, they cannot be used in the MANET directly because of the differences between the hardwired network and the MANET. Table 2.2 located after the protocols, gives the Characteristic comparison of proactive Unicast Routing Protocols.

**Optimized Link State Routing Protocol (OLSR):** Optimized Link State Routing Protocol (OLSR) (Jacquet 1998) is a proactive (table-driven) routing protocol for
MANETs. A route between source to destination is available immediately when needed. OLSR is based on the link-state algorithm. Conventionally, all wireless nodes flood neighbour information in a link-state protocol, but not in OLSR node. It is to advertise information only about links with a neighbour who is in its multipoint relay selector set. Its reduced size of control packets reduces flooding by using only multipoint relay nodes to send information in the network and reduce number of control packets by reducing duplicate transmission. This protocol does not expect reliable transfer, since updates are sent periodically. OLSR uses hop-by-hop routing. Routes are based on dynamic table entries maintained at intermediate nodes. The protocol is designed to work in distributed manners and thus does not depend on the central entity. The protocols thus support a nodal mobility that can be traced through its local control message, which depends up on the frequency of these messages. Advantage of OLSR is having the routes available within the standard routing table can be useful for some systems and network applications as there is no route discovery delay associated with finding a new route. Bigger overhead and necessity for more power are main disadvantage of this protocol.

**Fisheye State Routing Protocol (FSR):** The Fisheye State Routing (FSR) (Mario Gerla 2002) is a table driven unicast routing protocol for Mobile Ad hoc Networks based on Link State routing algorithm in effect with reduced overhead to keep network topology information. As showed in its name, FSR utilizes a function similar to a fish eye. The eyes of fishes catch the pixels near the focal with high detail, and the detail decreases as the distance from the focal point increases. Similar to fish eyes, FSR maintains the accurate distance and path quality information about the immediate neighbouring nodes, and progressively reduces detail as the distance increases. Advantage of this protocol is that it has potentiality to support multiple-
path routing and QoS routing but disadvantage of FSR is that it has high storage complexity.

**Topology Broadcast Based on Reverse-Path:** Forwarding Routing Protocol (TBRPF) Topology Broadcast Based on Reverse-Path Forwarding Routing Protocol (TBRPF) was proposed in 2002 (Richard G Ogier 2002). TBRPF aims at the Mobile Ad hoc Network with at most several hundreds of mobile nodes or high mobility of nodes. Every node in the wireless network keeps partial global topology information. When a node needs the shortest path to every other node, a minimum spanning tree rooted at itself is computed using modified Dijkstra’s algorithm. TBRPF transmits only the differences between the previous network state and the current network state. Therefore, routing messages are smaller, and can therefore be sent more frequently. This means that nodes’ routing tables are more up-to-date.

**Table 2.2 Characteristic of Proactive Unicast Routing Protocol**

<table>
<thead>
<tr>
<th></th>
<th>OLSR</th>
<th>FSR</th>
<th>TBRPF</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Scope</strong></td>
<td>Large and dense MANETs</td>
<td>Large scale MANETs with high mobility</td>
<td>MANETs with hundreds of nodes and high mobility</td>
</tr>
<tr>
<td><strong>Organization Of the network</strong></td>
<td>Flat</td>
<td>Hierarchical</td>
<td>Flat</td>
</tr>
<tr>
<td><strong>Neighbor Detection method</strong></td>
<td>Periodical HELLO messages</td>
<td>Periodical link state updates</td>
<td>Differential HELLO messages</td>
</tr>
<tr>
<td><strong>Optimized Broadcast</strong></td>
<td>Multipoint relaying</td>
<td>Combined with neighbor Detection</td>
<td>Combined with HELLO messages</td>
</tr>
<tr>
<td><strong>Broadcast Information</strong></td>
<td>MPR selector list</td>
<td>Link state update</td>
<td>(Partial) Spanning tree</td>
</tr>
<tr>
<td><strong>Route freshness</strong></td>
<td>Up-to-date</td>
<td>Maybe not up-to-date</td>
<td>Up-to-date</td>
</tr>
</tbody>
</table>
2.4.1.2 Reactive unicast routing protocols

Due to frequently changing topology of the Mobile Ad hoc Network, the global topology information stored at each node which consumes lots of bandwidth needs to be updated frequently. However, this consumption sometimes is a waste of bandwidth, because the link state updates received expires before the route between itself and another node is needed. To minimize the wastage of bandwidth, the concept of On Demand or reactive routing protocol is proposed by David B. Johnson, 1994. In On Demand protocols, the routing is divided into following two steps: first one is route discovery and second one is route maintenance. The most distinctive On Demand unicast routing protocols are Dynamic Source Routing (DSR) protocol, Ad Hoc On demand Distance Vector Routing (AODV) protocol and Temporally Ordered Routing Algorithm etc. Table 2.3 gives the Characteristic comparison of Reactive Unicast Routing Protocols.

**Dynamic Source Routing Protocol (DSR):** Dynamic Source Routing (DSR) (Johnson 1996) is an On Demand unicast routing protocol that utilizes source routing algorithm. In source routing algorithm, each data packet contains complete routing information to reach its dissemination. Additionally, in DSR each node uses caching technology to maintain route information that it has discovered. For example, the intermediate nodes cache the route towards the destination and backward to the source. Furthermore, because the data packet contains the source route in the header, the overhearing nodes are able to cache the route in its routing cache.
**Ad Hoc On-demand Distance Vector Routing Protocol (AODV):** The Ad Hoc On-demand Distance Vector Routing (AODV) protocol (Perkins 1999) is a reactive unicast routing protocol for mobile ad hoc networks. As a reactive routing protocol, AODV only needs to maintain the routing information about the active paths. In AODV, routing information is maintained in routing tables at nodes. Every mobile node keeps a next-hop routing table, which contains the destinations to which it currently has a route. A routing table entry expires if it has not been used or reactivated for a pre-specified expiration time. Moreover, AODV adopts the destination sequence number technique used by DSDV in an on-demand way.

**Temporally Ordered Routing Algorithm:** Temporally Ordered Routing Algorithm (TORA) (Park 1997 and Corson 1997) is an On Demand routing algorithm based on the concept of link reversal. This Routing protocol improves the partial link reversal method by detecting partitions and stopping non-productive link reversals. TORA can be used for highly dynamic mobile ad hoc networks. TORA has three basic steps: route creation, route maintenance and route erasure. In TORA the DAG provides the capability that many nodes can send packets to a given destination and guarantees that all routes are loop-free. Because of node mobility the DAG in TORA may be disconnected. So, route maintenance step is a very important part of TORA. This routing protocol has the unique feature that control messages are localized into a small set of nodes near the topology changes occur.
Table 2.3 Characteristic of Reactive Unicast Routing Protocol

<table>
<thead>
<tr>
<th></th>
<th>DSR</th>
<th>AODV</th>
<th>TORA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Updating of Destination at Source</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multicast Capability</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Control Hello Message Requirement</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Design Structure</td>
<td>Flat</td>
<td>Flat</td>
<td>Flat</td>
</tr>
<tr>
<td>Unidirectional link</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Multiple Route</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

2.4.1.3 Hybrid unicast routing protocols

Hybrid routing protocol attempts to discover balance between the two such as proactive for neighbourhood and reactive for far away nodes. Based on proactive and reactive routing protocols, some hybrid routing protocols are proposed to combine their advantages. The most distinctive hybrid routing protocol is Zone Routing Protocol.

Zone Routing Protocol (ZRP): Zone Routing Protocol (ZRP) (Haas 1998) is a hybrid routing protocol for mobile ad hoc networks. The hybrid protocols are proposed to reduce the control overhead of proactive routing approaches and decrease the latency caused by route search operations in reactive routing approaches. Zone Routing Protocol (ZRP) is a framework of hybrid routing protocol suites, which is made up of the following modules: First one is Intra-zone Routing
Protocol, second one is Inter-zone Routing Protocol, and the last one is Border cast Resolution Protocol.

ZRP refers to the locally proactive routing component as the Intra-zone Routing Protocol (IARP). The globally reactive routing component is named Inter-zone Routing Protocol (IERP). IERP and IARP are not specific routing protocols. Instead, IARP is a family of limited-depth, proactive link-state routing protocols. IARP maintains routing information for nodes that are within the routing zone of the node. Correspondingly, IERP is a family of reactive routing protocols that offers enhanced route discovery and route maintenance services based on local connectivity monitored by IARP (Hass 2001).

2.4.2 Multicast Routing Protocols

Although multicast transmission has not been widely deployed in the current MANETs, it will become very important in multimedia communications in the near future. To send a same data packet to multiple receivers in the MANET simultaneously, the simplest method is to broadcast the data packets. However, broadcast consumes considerable bandwidth and power, which should be avoided as much as possible (Wu &Tay 1999). Multicast can be used for saving the bandwidth while transmitting same data packets to multiple receivers. Figure 2.4 shows the multicast process in which data packet is replicated by the network. There have been many multicast routing protocols proposed for MANET. They could be divided into three groups: first one is proactive multicast, second one is reactive multicast and the last one is hybrid multicast routing protocol.
2.4.2.1 Proactive multicast routing protocols

Conventional routing protocols such as Ad-hoc Multicast Routing (AMRoute), Core-Assisted Mesh Protocol (CAMP) and Ad-hoc Multicast Routing Protocol Utilizing Increasing id-numbers (AMRIS) are proactive multicast routing protocols. Periodic broadcast of network topology updates are needed to compute the shortest path from the source to every destination, which consumes a lot of bandwidth. Table 2.4 gives the Characteristic comparison of proactive Multicast Routing Protocol.

**Ad-hoc Multicast Routing (AMRoute):** Ad-hoc Multicast Routing (AMRoute) (Liu & Talpade) is a tree based multicast routing protocol for mobile ad hoc networks. AMRoute creates a multicast shared-tree over mesh. AMRoute relies on the existence of an underlying unicast routing protocol. AMRoute has two key phases: mesh creation and tree creation. This protocol can be used for networks in which only a set of nodes supports AMRoute routing function. It is only one logical core in the multicast tree, which is responsible for group member maintenance and
multicast tree creation. This routing protocol builds a user-multicast tree, in which only the group members are included; because non-members are not included in the tree, the links in the tree are virtual links. In other words, they are in fact multi-hop IP-in-IP tunnels and AMRoute depends on the underlying unicast routing protocol to deal with network dynamics, although it has no privilege for unicast routing protocols. AMRoute creates an efficient and robust shared tree for each group. It helps to keep the multicast delivery tree unchanged with changes of network topology, as long as paths between tree members and core nodes exist via mesh links. When mobility is present, AMRoute suffers from loop formation, creates non-optimal trees, and requires higher overhead to assign a new core. Also, AMRoute suffers from a single point of failure of the core node (Osamah2009).

**Ad hoc Multicast Routing Protocol utilizing increasing id-numbers (AMRIS):**

AMRIS (Wu 1999) is a proactive shared tree based multicast routing protocol, which is independent of the fundamental unicast routing protocol. In AMRIS, the tree maintenance procedure operates continuously and locally to ensure a node’s connection to the multicast session delivery tree. In AMRIS, the tree maintenance procedure operates continuously and locally to ensure a node’s connection to the multicast session delivery tree. AMRIS is an on-demand protocol that constructs a shared delivery tree to support multiple senders and receivers within a multicast session. AMRIS dynamically assigns every node (on demand) in a multicast session with an ID number known as msm-id. The msm-id provides a heuristic height to a node and the ranking order of msm-id numbers directs the flow of datagram in the multicast delivery tree. Every node calculates its msm-id during the initialization phase, which is initiated by a special node called S-id. Normally, the S-id is the source node if there is only one source for the session. Otherwise, the S-id is the source node that has the minimum msm-id. The S-id broadcasts a NEW_SESSION
message to its neighbours. When a node wants to join the multicast session, it chooses one of its neighbours which has the smaller msm-id as its parent and sends a JOIN-REQ message. If the neighbour is in the tree (if the tree has been built), it answers with a JOIN-ACK message, which means the connection is successful; otherwise (when it is the first time to build the tree), the neighbour forwards JOIN-REQ to its own neighbours and waits for the reply, which is repeated until the JOIN-REQ arrives at an on-tree node or the source. As a result, a delivery tree rooted from the source is formed to include all the group members and some relay non-members. AMRIS repairs the broken links by performing local route repair without the need for any central controlling node, thereby reducing the control overhead.

**Core-Assisted Mesh protocol (CAMP):** Core-Assisted Mesh Protocol (CAMP) (Garcia-Luna-Aceves 1999) is a proactive multicast routing protocol based on shared meshes. The mesh structure provides at least one path from each source to each receiver in the multicast group. CAMP relies on an underlying unicast protocol which can provide correct distances to all destinations within finite time. Every node maintains a Routing Table (RT) that is created by the underlying unicast routing protocol. CAMP modifies this table when a multicast group joins or leaves the network. A Multicast Routing Table (MRT) is based on the Routing Table that contains the set of known groups. Moreover, all member nodes maintain a set of caches that Sender contains previously seen data packet information and unacknowledged membership requests. The creation and maintenance of meshes are main parts of CAMP.
Table 2.4 Characteristic of Proactive Multicast Routing Protocol

<table>
<thead>
<tr>
<th>Structure of Multicast delivery</th>
<th>AMRoute</th>
<th>AMRIS</th>
<th>CAMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loop free</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Dependency on Unicast routing protocol</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Scalability</td>
<td>Fair</td>
<td>Fair</td>
<td>Good</td>
</tr>
<tr>
<td>Control Packet flooding</td>
<td>Flat</td>
<td>Flat</td>
<td>Flat</td>
</tr>
<tr>
<td>Periodic message Requirement</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

2.4.2.2 Reactive multicast routing protocols

Traditional routing protocols such as On-Demand Multicast Routing Protocol (ODMRP) and Multicast Ad hoc on-demand Distance Vector (MAODV) are Reactive multicast routing protocols. Reactive routing means discovering the route when needed. Reactive routing protocols are well suited for a large-scale, narrow-band MANET with moderate or low mobility. The Table 2.5 gives the Characteristic comparison of Reactive Multicast Routing Protocol.

**On-Demand Multicast Routing Protocol (ODMRP):** On-Demand Multicast Routing Protocol (ODMRP) (Lee1999) is a reactive mesh based multicast routing protocol. ODMRP is not only a multicast routing protocol, but also provides unicast routing capability. The source establishes and maintains group membership and multicast mesh on demand if it needs to send data packets to the multicast group,
which is somewhat similar to MAODV. A set of nodes, which is called forwarding
group, participates in forwarding data packets among group members. All the states
in ODMRP are soft states which are refreshed by the control messages mentioned
above or data packets which achieve higher robustness. ODMRP uses a forwarding
group concept for multicast packet transmission, in which each multicast group G is
associated with a Forwarding Group (FG). Nodes in FG are in charge of forwarding
multicast packets of group G. In a multicast group of ODMRP, the source manages
the group membership, establishes and updates the multicast routes on demand. Like
reactive unicast routing protocols, ODMRP comprises two main phases: the request
phase and the reply phase. When a multicast source has a packet to send but it has
no routing and group membership information, it floods a Join Request packet to the
entire network. Join Request packets are member-advertising packets with
piggybacked data payload. When a node receives a non-duplicate JOIN Request, it
stores the upstream node ID in its routing table and rebroadcasts the packet. When
the JOIN Request packet reaches a multicast receiver, the receiver refreshes or
creates an entry for the source in Member Table and broadcasts JOIN TABLE
packets periodically to its neighbours. When a node receives a JOIN TABLE packet,
it checks each entry of the table to find out if there is an entry in the table whose
next node ID field matches its ID. If there is a match, the node recognizes that it is
on the path to the source, thus it is part of the forwarding group. Then it sets the
FG_FLAG and broadcasts its own JOIN TABLE built upon matched entries.
Consequently, each member of a forwarding group propagates the JOIN TABLE
packets until the multicast source is reached via the shortest path. This process
constructs (or updates) the routes from sources to receivers and builds a mesh of nodes, which is the forwarding group.

**Multicast Ad-hoc On-demand Distance Vector (MAODV):** Multicast operation of Ad-hoc On-demand Distance Vector (MAODV) (Royer 1999) is a reactive tree-based multicast routing protocol. MAODV is an extension of the unicast routing protocol Ad-hoc On-demand Distance Vector (AODV). Using MAODV, all nodes in the network maintain local connectivity by broadcasting “Hello” messages with TTL set to one. Every node maintains three tables, a Routing Table (RT), a Multicast Routing Table (MRT) and a Request Table. RT stores routing information and has the same function as in AODV. In unicast routing operations, every destination has a unique sequence number. Likewise, every multicast group has a sequence number to indicate the freshness of the multicast routing information. Thus, one and only one group leader is elected to broadcast periodical GROUP HELLO messages throughout the MANET to maintain the sequence number. The group leader is by default the first node joining the group, but could be another node when the first node leaves the group. The main drawbacks of MAODV are long delays and high overheads associated with fixing broken links in conditions of high mobility and traffic load. Also, it has a low packet delivery ratio in scenarios with high mobility, large numbers of members, or a high traffic load. Because of its dependence on AODV, MAODV is not flexible. Finally, it suffers from a single point of failure, which is the multicast group leader. (Osamah2009)
Table 2.5 Characteristic of Reactive Multicast Routing Protocol

<table>
<thead>
<tr>
<th></th>
<th>ODMRP</th>
<th>MAODV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multicast delivery</td>
<td>Mesh</td>
<td>Core based tree</td>
</tr>
<tr>
<td>structure</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Loop free</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Periodic messages</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>requirement</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Routing Hierarchy</td>
<td>Flat</td>
<td>Flat</td>
</tr>
<tr>
<td>Scalability</td>
<td>Flat</td>
<td>Fair</td>
</tr>
</tbody>
</table>

2.4.2.3 Hybrid Multicast Routing Protocols

Traditional routing protocol such as Optimized Polymorphic Hybrid Multicast Routing Protocol (OPHMR) is the Hybrid multicast routing protocol. Hybrid routing protocol attempts to discover balance between the two such as proactive for neighbourhood and reactive for far away nodes.

**Optimized Polymorphic Hybrid Multicast Routing Protocol (OPHMR):** This protocol (Mnaouer 2007) is invested with different operational modes that are either proactive or reactive based on a MN’s power remainder, mobility level, and vicinity density level. It attempts to address the issues of power efficiency, latency, and protocol overhead in an adaptive manner. OPHMR’s reactive behaviour is based on the On-Demand Multicast Routing Protocol (ODMRP). It is relatively simplistic. It generates on-demand route paths for multicast message requests. OPHMR’s proactive behaviour is based on the Multicast Zone Routing (MZR) protocol. It builds a zone around each Mobile Node (in hops) and periodically sends updates within each defined zone. For added efficiency, OPHMR utilizes an optimizing scheme adapted from the Optimized Link State Routing (OLSR) protocol. It used to
decrease the amount of control overhead that is produced. OPHMR is, after a very lengthy period of time, able to extend battery life and enhance the survivability of the mobile ad hoc nodes. As a result, it decreases the end-to-end delay and increases the packet delivery ratio.

2.5 MANET SIMULATION AND SIMULATORS

Simulation is an often used tool to analyze MANETs, illustrated by the fact that 84 out of the 111 Mobile Ad Hoc Networking and Computing (MobiHoc) papers published (75.7%) used simulation to present research results (Stuart Kurkowski). As the MANET community moves forward toward implementation, simulation research becomes imperative.

Figure 2.5 shows the simulator usage results of the MobiHoc authors that did identify the simulator used. Table 2.6 provides an estimation of the popularity of simulators, their programming language, and the license they use (Luc Hogie 2006). As there is no statistics available on simulators users, the number of web-pages that refer to each simulator is considered. This gives an order of magnitude of their popularity.

Table 2.6 Simulation Popularity, License and Programming.

<table>
<thead>
<tr>
<th>Name</th>
<th>Popularity</th>
<th>License</th>
<th>Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>ns-2</td>
<td>88.8%</td>
<td>Open Source</td>
<td>C++/OTCL</td>
</tr>
<tr>
<td>Glomosim</td>
<td>4%</td>
<td>Open Source</td>
<td>Parsec (C-based)</td>
</tr>
<tr>
<td>OPNet</td>
<td>2.61%</td>
<td>Commercial</td>
<td>C</td>
</tr>
<tr>
<td>OMNET++</td>
<td>1.04%</td>
<td>Free for academic use</td>
<td>C++</td>
</tr>
<tr>
<td>J-Sim</td>
<td>0.45%</td>
<td>Open source</td>
<td>Java</td>
</tr>
</tbody>
</table>
Figure 2.5 Simulator usage results.

MANET simulation could be categorized into four stages namely simulation setup, simulation execution, output analysis, and publishing. There exist two techniques which provide feedback on what happens within the simulation. Discrete-event simulation encourages the generation of a trace file containing a description of each event that occurred. This general technique reports all events to the user. Unfortunately because of the huge size of the generated trace, this technique consumes a lot of CPU resource and thus significantly slows down the simulation process. Finally, the user needs to write text-processing scripts (called filters in the Unix world) in order to retain only what he is interested in and to generate GNU Plot files.

The alternative consists in dynamically interacting with the simulation engine by resorting to the observer design pattern. The measurement system is then event-driven. More precisely, the user initially announces the classes of events he is interested in. He will then be dynamically notified of such events as the simulation process progresses.
MANETs simulators exhibit different features and models. The choice of a simulator should be driven by the requirements. Determining the level of details required is key. If high-precision PHY layers are needed, then ns-2 is clearly the wisest choice. On the contrary, if the wireless technology has no impact on the targeted protocol, recent simulators which propose high-level abstractions and polished object-oriented designs will be adapted.

The number of nodes targeted determines the choice of the simulation tool. Sequential simulators should not be expected to run more than 1,000 nodes. If larger scales are needed, then parallel simulators are a wise choice. Highly optimized simulators like ns-2 coupled with stage simulation may be considered. Finally, most non-commercial simulators suffer from a lack of good documentation and support. Using a commercial one might help in case of troubles. Moreover, commercial simulators usually feature extensive lists of supported protocols, while open source solutions give full empowerment.

Because of the complex nature of the MANETs, their simulation is a very challenging issue. Simulators rely on various techniques for improving their accuracy, speed, scalability, usability, etc. As the needs of researchers continue to evolve, it is likely that existing simulators will integrate new functionalities and concepts as well as fresh simulators will be developed. Several techniques have been proposed to overcome the simulation pitfalls in every stage of the simulation process. New simulation tools and updated versions of the existing tools are released to overcome these pitfalls thus making MANET simulators to be an integral part of MANET research community.
2.5.1 MANETs Simulation Techniques

Simulation can be either continuous or discrete. Continuous simulation makes use of analytic models. Because of the intrinsic complexity of the MANETs, analytic models can hardly be applied. Discrete simulation proves to be more practicable. In the case of MANETs simulation, discrete simulation can benefit from specific optimization techniques. This section gives an overview of general and such MANETs-specific techniques implemented within simulators. Table 2.7 details the runtime properties for each simulator.

2.5.1.1 Parallelism and distribution

Parallelism refers to the simultaneous execution of different instructions of the same program. It is used to quicken simulations. Parallelism is a relevant technology for simulating wired networks. Distribution refers to the repartition of program data or code (or both) on distinct computers. It is primarily used to bring scalability and/or to enable parallelism. Parallelism and distribution can be coupled or used independently. The improvement in terms of quickness/scalability depends on the number of processors/nodes involved. When distribution is not used, parallelism is typically applied on shared memory architectures (SMP). This technique allows simulators to model networks made of tens of thousands stations. It is implemented by GloMoSim (Xiang Zeng 1998). Distribution benefits from the recent interest in Beowulf clusters (local computational grids made of numerous low-cost workstations, typically PC/Linux boxes). Parallelism and Distribution relies on this technique which allows it to simulate hundreds of thousands stations. The COMPASS (http://www.cc.gatech.edu/computing/compass/) project at GeorgiaTech, which makes an extensive use of distribution by building
heterogeneous distributed simulations consisting of instances of ns-2 and GloMoSim operating together.

**Table 2.7 Parallelism and Interface of Simulators**

<table>
<thead>
<tr>
<th>Name</th>
<th>Parallelism</th>
<th>Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>ns-2</td>
<td>No</td>
<td>C++/OTCL</td>
</tr>
<tr>
<td>DIANEmu</td>
<td>No</td>
<td>Java</td>
</tr>
<tr>
<td>Glomosim</td>
<td>SMP/Beowulf</td>
<td>Parsec (C-based)</td>
</tr>
<tr>
<td>GTNets</td>
<td>SMP/Beowulf</td>
<td>C++</td>
</tr>
<tr>
<td>J-Sim</td>
<td>RMI-based</td>
<td>Java</td>
</tr>
<tr>
<td>Jane</td>
<td>No</td>
<td>Java</td>
</tr>
<tr>
<td>NAB</td>
<td>No</td>
<td>OCaml</td>
</tr>
<tr>
<td>OMNet++</td>
<td>MPI/PVM</td>
<td>C++</td>
</tr>
<tr>
<td>OPNet</td>
<td>Yes</td>
<td>C</td>
</tr>
<tr>
<td>Pdns</td>
<td>Beowulf</td>
<td>C++/OTCL</td>
</tr>
<tr>
<td>QualNet</td>
<td>SMP/Beowulf</td>
<td>Parsec (C-based)</td>
</tr>
<tr>
<td>SWANS</td>
<td>No</td>
<td>Java</td>
</tr>
</tbody>
</table>

**2.5.1.2 Staged simulation**

Staged simulation is described by Walsh and Sirer (KevinWalsh 2004). It is a general technique which improves the performance of discrete-event simulators by identifying and eliminating redundant computations. It consists of three parts: function caching, event-restructuring, and time-shifting. Function caching avoids redundant computations by placing into a memory cache the arguments and results of function calls. Event-restructuring improves on function caching by exposing low-level events that otherwise would have not been treated by function caching. Time-shifting reorders the events into a sequence that is better suited to the
computer architecture that executes the simulator. It enables a sequence of small, consecutive events to be computed altogether by a single and more efficient algorithm. The SNS project applies staged simulation to ns-2 and boasts a runtime 30 times faster than the original one.

2.5.1.3 Bining

Bining makes good use of the spatial localization of network nodes in the MANETs—it is actually practicable and widely employed in all systems made of spatially located objects. It consists of the simulation area into a list-based (applicable on 1-dimensional simulation space), grid-based (also referred to as flat bining, applicable on 2D spaces) or tree-based (also referred to as flat bining, applicable on 3D spaces) structure. Bining dramatically improves the determination of the communication links in the network. For example, applying flat bining reduces the complexity of this process from \( O(n^2) \) to \( O(n) \) (\( n \) being the number of stations in the network). This technique is described by Naoumov and Gross (Valeri Naoumov 2003), who applied it to ns-2.

2.5.1.4 Hybrid simulations

The principle of hybrid simulations is to mix analytic models and discrete ones. Although pure analytic models (Matthias Scheidegger 2005) do not suit MANETs simulation, mixing them with discrete model leads to good results. Hybrid simulation is being investigated and applied to MANETs simulation by Lu and Schormans (Shaowen Lu 2003).

2.5.2 MANET Simulators

This section presents the list of available MANET Simulators in detail.
**DIANEmu**: (Klein2003) is a discrete-event simulator developed at Karlsruhe University (Germany). It aims to enable the simulation of ad hoc applications in realistic contexts. So far, most simulators have been designed to permit simulations at a protocol-level. DIANEmu’s approach is different: it assumes that the lowest network layers (up to the fourth one) are available. DIANEmu then focuses on the application model. DIANEmu belongs to a new class of simulators which allow the large-scale simulation of high-level applications such as gaming and e-business.

DIANEmu provides a complete environment for application design. Its simulation engine is closely coupled to its graphical interface. Attesting of its modern design, its measurement system is event-driven. More precisely it defines that to each event class is associated to a given handler (referred to as a gauge). This handler is then dynamically invoked when the events of the specified class occur. DIANEmu is written in Java and is free.

**GloMoSim**: (Xiang Zeng 1998) is developed at UCLA (California, USA). It is the second most popular wireless network simulator. GloMoSim is written in Parsec and hence benefits from the latter’s ability to run on shared-memory symmetric processor (SMP) computers. New protocols and modules for GloMoSim must be written in Parsec too. GloMoSim respects the OSI standard. The parallelization technique used by GloMoSim is the same as the pdns’s one; that is the network is split in different sub-networks, each of them being simulated by distinct processors. The network is partitioned in such a way that the number of nodes simulated by each partition is homogeneous.

GloMoSim uses a message-based approach to discrete-event simulation. More precisely, network layers are represented as objects called entities. Events are
represented as time-stamped messages handled by entities. GloMoSim’s network model does not define every network nodes as entities because this would lead to too numerous objects. Instead, GloMoSim uses entities to model network layers. Messages - which represent network events—then cross the layer stack by being interchanged by the entities. GloMoSim can simulate networks made up of tens of thousands devices.

Just like ns-2, realistic simulations have been made possible by extensions such as the Obstacle mobility model (Amit Jardosh 2005) and the GEMM project (Ray 2004). A java-based visualization tool is provided. The qualities of GloMoSim permitted it to be chosen as the core of the commercial QualNet simulator (detailed hereinafter). Although it is a good tool for MANETs simulation, GloMoSim suffers from a lack of a good and in-depth documentation.

GTNets : GTNets (George F Riley 2003) is developed at GeorgiaTech institute (Atlanta, USA). According to its authors, the design philosophy of GTNetS is to create a simulation environment that is structured much like actual networks are structured. More precisely, in GTNetS, there is clear and distinct separation of protocol stack layers and the network programming interface used by applications use function calls similar to the ubiquitous POSIX standard. The parallelization ability of GTNetS makes it possible to distribute a single simulation over a network of loosely coupled workstations, a shared-memory symmetric multiprocessing system (SMP), or a combination of both. This endows GTNetS with good scalability and then allows the simulation of large networks. Concerning the support of protocols, IEEE802.11 as well as Bluetooth (Xin Zhang 2004) are implemented. Another benefit of GTNets is that the simulator gathers statistics regarding its own performance. The graphical user interface provided with GTNetS supports the
graphical representation of the simulation topology, with selective enabling and disabling of display for specified nodes and links. It is an open source.

**J-Sim:** ([http://javasim.ncl.ac.uk](http://javasim.ncl.ac.uk)) (formerly known as JavaSim) is developed at Ohio and Illinois Universities (USA). It is a component-based, compositional simulation environment. Initially designed for wired network simulation, its Wireless Extension proposes an implementation of the IEEE802.11 MAC—which is the only MAC supported so far. This extension turns J-Sim to a viable MANETs simulator. J-Sim also features a set of components which facilitates basic studies of wireless/mobile networks, including three distinct radio propagation models and two stochastic mobility models. J-Sim is written in Java and is open source.

**Jane:** Hannes Frey (2003), Johannes K Lehnert (2004) is developed at Trier University (Germany). It consists of both a simulation environment and an execution platform. Its main interesting feature is that it allows the simulation code to be migrated to the real devices without any modification. Jane also features an emulation mode that allows real devices to participate in simulations. In addition to that, Jane features high level concepts (such as the notions of service, message, etc) that are suitable to the simulation of applications-level services. It makes use of GPS information, which turns it to an appealing tool for the simulation location-based services. Jane is written in Java and is an open source.

**NAB:** EPFL (2004) (Network in A Box) is a discrete event simulator developed at EPFL (Lausanne, Switzerland). NAB is dedicated to MANET’s simulation. By focusing on scalability and visualization and by featuring a very realistic mobility model (a constrained waypoint based on city maps), it meets the needs of cutting-edge applications. According to its author, Nab was born out of the
inability to simulate large ad hoc networks with existing tools, and some impatience in dealing with their internal complexity which tended to make implementing new functionality a lengthy and bug-ridden task. NAB’s design is node-oriented (and object-oriented); that is, each node is represented by an object. It is written in OCaml and is actually the only simulator written in a language whose syntax is not derived from C. It is an open source.

**NS-2**: Network Simulator (1995) is the de facto standard for network simulation. Its behaviour is highly trusted within the networking community. It is developed at ISI, California, and is supported by the DARPA and NSF. NS-2 is a discrete-event simulator organized according to the OSI model and primarily designed to simulate wired networks.

The support for wireless networking had been brought by several extensions. The Monarch CMU projects (http://www.monarch.cs.cmu.edu/) made available an implementation of the IEEE802.11 layers (WiFi). The BlueHoc and BlueWare (Rimon Barr 2002) projects provided the Bluetooth layers.

NS-2 provides a set of randomized mobility models, including random waypoint. Advanced node mobility had been made available by the Graph Mobility project (Jing Tian, 2002), the GEMM project (Ray 2004), and the Obstacle Mobility (Amit Jardosh 2003 and 2005) model. These constitute a progress towards realistic simulation.

The core of ns-2 is a monolithic piece of C++ code. It is extendable by adding C++ modules. The configuration relies on OTCL (a dialect of TCL by MIT) scripts. ns-2 then appears to the user as an OTCL interpreter. More precisely, it reads scenarios files written in OTCL and produces a trace file in its own format. This trace needs to be processed by user scripts or converted and rendered using the
NAM tool. Thanks to its open source licence and its popularity, that new extensions are sporadically proposed. For example, Dricot and De Doncker (Dricot 2004) proposed a highly accurate physical model based on ray tracing and Markov chains. This extension, which can be very useful for MANETs simulation, makes the simulator to be about 100 times slower.

ns-2 is a sound solution to MANET simulation. Unfortunately it suffers because of its lack of modularity and its inherent complexity (ns-2 was candidate to be the basis for the Qualnet (http://www.scalable-networks.com/products/qualnet.php) simulator but got finally rejected). Indeed, adding components/protocols or modifying existing ones is not as straight forward as it should be. Ns-2 is said to have few good documentation for quite long time. Tutorials or example driven documentations posted online by several users have changed this scenario recently.

Another well-known weakness of ns-2 is its high consumption of computational resources. A harmful consequence is that ns-2 lacks scalability which impedes the simulation of large networks (ns-2 is typically used for simulations consisting of no more than a few hundred nodes). Several projects have aimed at improving ns-2’s runtime. For example, staged simulation (Kevin Walsh 2004) and parallelism have turned out to be efficient solutions.

**Pdns**: George F Riley (1999) is developed at Gorgia Tech institute, California. The Parallel/Distributed Network Simulator aims at overcoming the limitation of ns-2 regarding its scalability. Pdns boosts ns-2 processes by distributing the simulation over a network of closely coupled workstations (a common TCP/IP is usable). More precisely, it achieves an efficient parallelization of the simulation process by making distinct instances of ns-2 simulating distinct sub-networks. Pdns can simulate networks consisting of up to hundreds of thousands nodes.
OMNet++: Imre (2001) is a well-designed simulation package written in C++. OMNET++ is actually a general-purpose simulator capable of simulating any system composed of devices interacting with each other. It can then perfectly be used for MANETs simulation. The mobility extension for OMNeT++ (Drytkiewicz 2003) is intended to support wireless and mobile simulations within OMNeT++. This support is said to be fairly incomplete. OMNet++ is for academic and educational use.

OPNet: Desbrandes (1993) (Optimized Network Engineering Tools) is a discrete-event network simulator first proposed by MIT in 1986. It is a well-established and professional commercial suite for network simulation. It is actually the most widely used commercial simulation environment. OPNET Modeler features an interactive development environment allowing the design and study of networks, devices, protocols, and applications. For this, extensive lists of protocols are offered. Particularly, MAC protocols include IEEE802.11a/b/g and Bluetooth ones. One of the most interesting features of OPNet is its ability to execute and monitor several scenarios in a concurrent manner. In spite of its wide adoption, some doubts remain regarding the dependability of its MANETs simulation engine. More precisely, Cavin and al. (David Cavin 2002) simulated a broadcasting process on the OPNet, ns-2 and GloMoSim simulators. It showed that the results obtained using OPNet were barely comparable to those harvested out of ns-2 and GloMoSim which exhibited similar behaviours. The divergences were quantitative but also qualitative (not the same general behaviour). OPNet is written in C++.

QualNet: This (http://www.scalable-networks.com/products/qualnet.php) is a commercial ad hoc network simulator based on the GloMoSim core. It extends the GloMoSim offer by bringing support, a decent documentation, a complete set of user-friendly tools for building scenarios and analyzing simulation
output. QualNet also largely extends the set of models and protocols supported by the initial GloMoSim distribution. As it is built on top of GloMoSim, QualNet is written in Parsec (Rajive Bagrodia 1998).

**SWANS**: This is (Rimon Barr 2004) developed at Cornell university, is a Java-based wireless network simulator built atop the JiST discrete event platform. SWANS boasts a highly efficient sequential simulation engine and has been compared to GloMoSiM, in terms of quality. JiST relies on the concept of virtual-machine simulation. The way SWANS implements simulated time is singular: the simulated time is not managed by some shared clock. Instead, each entity (referred to as a TimeFull entity) is in charge of determining the time needed to its execution. By invoking each TimeFull entities in a sequence, SWANS is able to determine the current simulated time. SWANS appears to the user as a framework. It must be programmed in plain Java, using some specific programming interface. SWANS is developed in Java and is an open source.

### 2.5.3 Selection of Simulators

MANETs simulators exhibit different features and models. The choice of a simulator should be driven by the requirements. Determining the level of details required is the key. If high-precision PHY layers are needed, then ns-2 (coupled with the highly-accurate PHY) is clearly the wisest choice. On the contrary, if the wireless technology has no impact on the targeted protocol, recent simulators (like NAB or Jane) which propose high-level abstractions and polished object-oriented designs will be more adapted.

The number of nodes targeted determines the choice of the simulation tool. Sequential simulators should not be expected to run more than 1,000 nodes. If larger scales are needed, then parallel simulators are a wise choice. It may be
considered highly optimized simulators like ns-2 coupled with stage simulation. Finally, most non-commercial simulators suffer from a lack of good documentation and support. Using a commercial one might help in case of troubles. Moreover, commercial simulators usually feature extensive lists of supported protocols, while open source solutions give full empowerment.

2.6 CONCLUSION

This chapter has given a detailed description of MANET Protocols, Performance Metrics and Simulators. A meticulous survey done on existing routing protocols and their characteristics is elaborated. This chapter has presented a set of challenges and problems encountered by MANETS. Hence, the ensuing chapters present the proposed strategies and routing techniques that cater to the improvement of overall performance of the MANET routing by tuning rerouting time as the performance metric.