Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2009, Article ID 764047, 42 pages
doi:10.1155/2009/764047

Review Article

Multicast Routing Protocols in Mobile Ad Hoc Networks: A Comparative Survey and Taxonomy

Osamah S. Badarneh and Michel Kadoch

Départment de Génie Electrique, École de Technologie Supérieure, Université du Québec, Montreal, Canada H3C 1k3

Correspondence should be addressed to Osamah S. Badarneh, osamah.badarneh.1@ens.etsmtl.ca

Received 14 January 2009; Accepted 14 June 2009

Recommended by Sudip Misra

Multicasting plays a crucial role in many applications of mobile ad hoc networks (MANETs). It can significantly improve the performance of these networks, the channel capacity (in mobile ad hoc networks, especially single-channel ones, *capacity* is a more appropriate term than *bandwidth*, capacity is measured in bits/s and bandwidth in Hz) and battery power of which are limited. In the past couple of years, a number of multicast routing protocols have been proposed. In spite of being designed for the same networks, these protocols are based on different design principles and have different functional features when they are applied to the multicast problem. This paper presents a coherent survey of existing multicasting solutions for MANETs. It presents various classifications of the current multicast routing protocols, discusses their operational features, along with their advantages and limitations, and provides a comparison of their characteristics according to several distinct features and performance parameters. Moreover, this paper proposes classifying the existing multicast protocols into three categories according to their layer of operation, namely, the network layer, the application layer, and the MAC layer. It also extends the existing classification system and presents a comparison between them.

1. Introduction

Mobile ad hoc networks (MANETs) comprise either fixed or mobile nodes connected wirelessly without the support of any fixed infrastructure or central administration. The nodes are self-organized and can be deployed "on the fly" anywhere, any time to support a particular purpose. Two nodes can communicate if they are within each other's transmission range; otherwise, intermediate nodes can serve as relays (routers) if they are out of range (multihop routing). These networks have several salient features: rapid deployment, robustness, flexibility, inherent mobility support, highly dynamic network topology (device mobility, changing properties of the wireless channel, that is, fading, multipath propagation, and partitioning and merging of ad hoc networks are possible), the limited battery power of mobile devices, limited capacity, and asymmetric/unidirectional links. MANETs are envisioned to support advanced applications such as military operations (formations of soldiers, tanks, planes), civil applications (e.g., audio and video conferencing, sport events, telematics applications (traffic)), disaster situations (e.g., emergency and rescue operations, national crises, earthquakes, fires, floods), and integration with cellular systems [1–3].

Multicasting plays a crucial role in MANETs to support the above applications. It involves the transmission of a datagram to a group of zero or more hosts identified by a single destination address, and so is intended for group-oriented computing. A multicast datagram is delivered to all members of its destination host group with the same "best effort" reliability as regular unicast IP datagrams, that is, the datagram is not guaranteed to arrive intact at the destinations of all members of the group, or in the same order relative to other datagrams [4]. The use of multicasting within MANETs has many benefits. It can reduce the cost of communication and improve the efficiency of the wireless channel when sending multiple copies of the same data by exploiting the inherent broadcasting properties of wireless transmission. Instead of sending data via multiple unicasts, multicasting minimizes channel capacity consumption, sender and router processing, energy consumption, and delivery delay, which are considered important MANET

factors. In addition, multicasting provides a simple yet robust communication method whereby a receiver's individual address remains unknown to the transmitter or changeable in a transparent manner by the transmitter [5, 6].

Multicasting in MANETs is much more complex than in wired networks and faces several challenges. Multicast group members move, which precludes the use of a fixed infrastructure multicast topology, wireless channel characteristics can vary over time, and there are restrictions on node energy and capacity [7]. The multicast protocols proposed for wired networks cannot be directly ported to MANETs due to the lack of mechanisms available for handling the frequent link breakages and route changes, or due to the differing characteristics of the two networks. Chiang et al. has proposed many mechanisms for adapting the wired multicast protocols to MANETs [8–11]. Simulation results in [8–11] show an increase in control packet overhead and a rapid decrease in throughput with increased node mobility. In addition, the simulation results show that these approaches indicate the need to explore alternative multicast strategies.

Several multicast routing protocols for MANETs have been proposed and evaluated [8–10, 12–44]. These protocols are based on different design principles and have different operational features when they are applied to the multicast problem. The properties favored depend on the protocol.

This paper is organized as follows: Section 2 presents the main issues and challenges that multicast protocols must address for adaptation to MANETs. Section 3 gives various classifications of existing multicast routing protocols in MANETs and describes their characteristics. Section 4 explains the multicast session life cycle. The functionality of some existing multicast routing protocols is presented in Section 5. Section 6 presents various criteria for evaluating the multicast routing protocols. Section 7 summarizes and compares the multicast protocols in a qualitative manner. Finally, Section 8 concludes the paper.

2. Multicast Routing Protocol Design: Issues and Challenges

The particular features of MANETs make the design of a multicast routing protocol a challenging one. These protocols must deal with a number of issues, including, but not limited to, high dynamic topology, limited and variable capacity, limited energy resources, a high bit error rate, a multihop topology, and the hidden terminal problem. The requirements of existing and future multicast routing protocols and the issues associated with these protocols that should be taken into consideration are listed in what follows [2, 3, 6, 45].

(i) Topology, Mobility, and Robustness. In MANETs, nodes are free to move anywhere, any time, and at different speeds. The random and continued movement of the nodes leads to a highly dynamic topology, especially in a high-mobility environment. A multicast routing protocol should be robust enough to react quickly with the mobility of the nodes and should adapt to topological changes in order to avoid dropping a data packet during the multicast session, which would create a low packet delivery ratio (PDR: the number of nonduplicate data packets successfully delivered to each destination versus the number of data packets supposed to be received at each destination). It is very important to minimize control overhead while creating and maintaining the multicast group topology, especially in an environment with limited capacity.

(ii) Capacity and Efficiency. Unlike wired networks, MANETs are characterized by scant capacity caused by the noise and interference inherent in wireless transmission and multipath fading. Efficient multicast routing protocols are expected to provide a fair number of control packets transmitted through the network relative to the number of data packets reaching their destination intact, and methods to improve and increase the available capacity need to be considered.

(iii) Energy Consumption. Energy efficiency is an important consideration in such an environment. Nodes in MANETs rely on limited battery power for their energy. Energy-saving techniques aimed at minimizing the total power consumption of all nodes in the multicast group (minimize the number of nodes used to establish multicast connectivity, minimize the number of overhead controls, etc.) and at maximizing the multicast life span should be considered.

(iv) Quality of Service and Resource Management. Providing quality of service (QoS) assurance is one of the greatest challenges in designing algorithms for MANET multicasts. Multicast routing protocols should be able to reserve different network resources to achieve QoS requirements such as, capacity, delay, delay jitter, and packet loss. It is very difficult to meet all QoS requirements at the same time because of the peculiarities of ad hoc networks. Even if this is done, the protocol will be very complex (many routing tables, high control overhead, high energy consumption, etc.). As a result, doing so will not be suitable for these networks with their scarce resources, and resource management and adaptive QoS methods are more convenient than reservation methods for MANETs.

(v) Security and Reliability. Security provisioning is a crucial issue in MANET multicasting due to the broadcast nature of this type of network, the existence of a wireless medium, and the lack of any centralized infrastructure. This makes MANETs vulnerable to eavesdropping, interference, spoofing, and so forth. Multicast routing protocols should take this into account, especially in some applications such as military (battlefield) operations, national crises, and emergency operations. Reliability is particularly important in multicasting, especially in these applications, and it becomes more difficult to deliver reliable data to group members whose topology varies. A reliable multicasting design depends on the answers of the following three questions [46]. By whom are the errors detected? How are error messages signaled? How are missing packets retransmitted?

(vi) Scalability. A multicast routing protocol should be able to provide an acceptable level of service in a network with a large number of nodes. It is very important to take into account the nondeterministic characteristics (power and capacity limitations, random mobility, etc.) of the MANET environment in coping with this issue.

3. Taxonomy of Multicast Routing Protocols

MANET multicast routing protocols can be classified into various categories [2, 45, 47–50]. We propose to classify the existing multicast protocols into three categories, according to their layer of operation, namely, the network layer, the application layer, and the MAC layer. In spite of being designed for the same type of underlying network, the characteristics of these two multicast routing protocols are quite distinct. The following sections describe these protocols and categorize them according to their characteristics. Figure 1 shows the various classifications of the multicast routing protocols in MANETs. This survey provides several advantages over other surveys.

(1) It classifies the existing multicast protocols according to their layer of operation, namely, the network layer, the application layer, and the MAC layer. We present the advantages and limitations of each layer with respect to its multicast. This will provide researchers with new ideas for designing new multicast protocols which take into consideration the advantages and limitations of each layer (cross-layer design).

(2) Previous surveys classify these protocols according to the popular classification methods (tree-based, mesh-based and/or proactive, and reactive). This survey presents comprehensive classifications of these protocols. We categorize them according to various features, such as layer of operation, routing mechanism, network topology, establishment of multicast connectivity, routing approach, and multicast group maintenance. In addition, each category is divided into a number of subcategories. Furthermore, the advantages and disadvantages of each category and subcategory are presented.

(3) It covers a huge number of multicast protocols and provides a comprehensive discussion of their operational features, along with their advantages and limitations, and provides a comparison of their characteristics according to several distinct feature and performance parameters. In addition, the operation of each protocol is portrayed diagrammatically, which helps us visualize the protocol mechanism.

(4) New ideas for future research are proposed, for example, interoperability, interaction, heterogeneity, and integration.

(5) It will serve as a quick reference guide, and provide readers with a comprehensive understanding of the design principles and the conceptual operations of multicast routing protocols.

3.1. Network Layer Multicasting versus Application Layer Multicasting versus MAC Layer Multicasting.

Multicasting protocols can be implemented at different layers of the protocol stack, such as the network layer (IP), the MAC layer, and the application layer, each of which can perform specific functions for supporting multicast communication. The network layer is responsible for routing data between a source-destination pair (end-to-end), while the MAC layer is responsible for ensuring that the data are correctly delivered to the destination (reliability), which requires the application layer to buffer data locally until acknowledgments (ACKs) have been received. However, it is the responsibility of the MAC layer to support rate adaptive multicasting.

Network Layer Multicast (IPLM). MANET multicasting has received a great deal of attention in terms of designing efficient protocols at the network (IP) layer [9, 12–15, 19, 20, 22–24, 26–30, 32–35, 39, 40, 42–44]. Protocols in this layer require the cooperation of all the nodes of the network. They also require forwarders (intermediate) nodes to maintain their pergroup state. The network (IP) layer implements minimal functionality, "best effort" unicast datagram service, while the overlay network implements multicast functionalities such as dynamic membership maintenance, packet duplication, and multicast routing.

Application Layer Multicast (ALM). ALM, or overlay multicast, has received little attention in the MANET domain [16, 18, 25, 36, 51]. Despite the fact that network layer multicast is known as the most efficient way to support multicast (since the majority of the proposed multicast protocols are implemented at the network layer), the overlay multicast handles several features, such as the following: (1) it is simple to deploy, because it does not require changes at the network layer; (2) intermediate (forwarder) nodes do not have to maintain their pergroup state for each multicast group (maintaining that state has always been a problem in multicasting, even on the Internet); (3) the creation of a virtual (logical) topology hides routing complications, such as link failure instances, which are left to be taken care of at the network layer; and finally (4) overlay multicasting can deploy the capabilities of lower-layer protocols in providing flow control, congestion control, security, or reliability according to the requirements of the application. For example, if the application needs reliability, it can choose, at run time, to use TCP between group members, and UDP, otherwise. Moreover, secure group communications are reduced to secure unicast communications, which makes it possible to avoid the use of complex protocols for the group key [18, 52].

The main problems with the overlay multicast method are routing efficiency and robustness. The robustness problem we refer to is that the distribution of the multicast tree is dependent on the end nodes. The routing efficiency we refer to is that the use of overlay multicasting can result in the transmission of multiple copies of multicast data packets over each physical link (multiple unicasts), which occurs because nonmulticast group members cannot make copies of multicast data packets. This effect clearly appears when

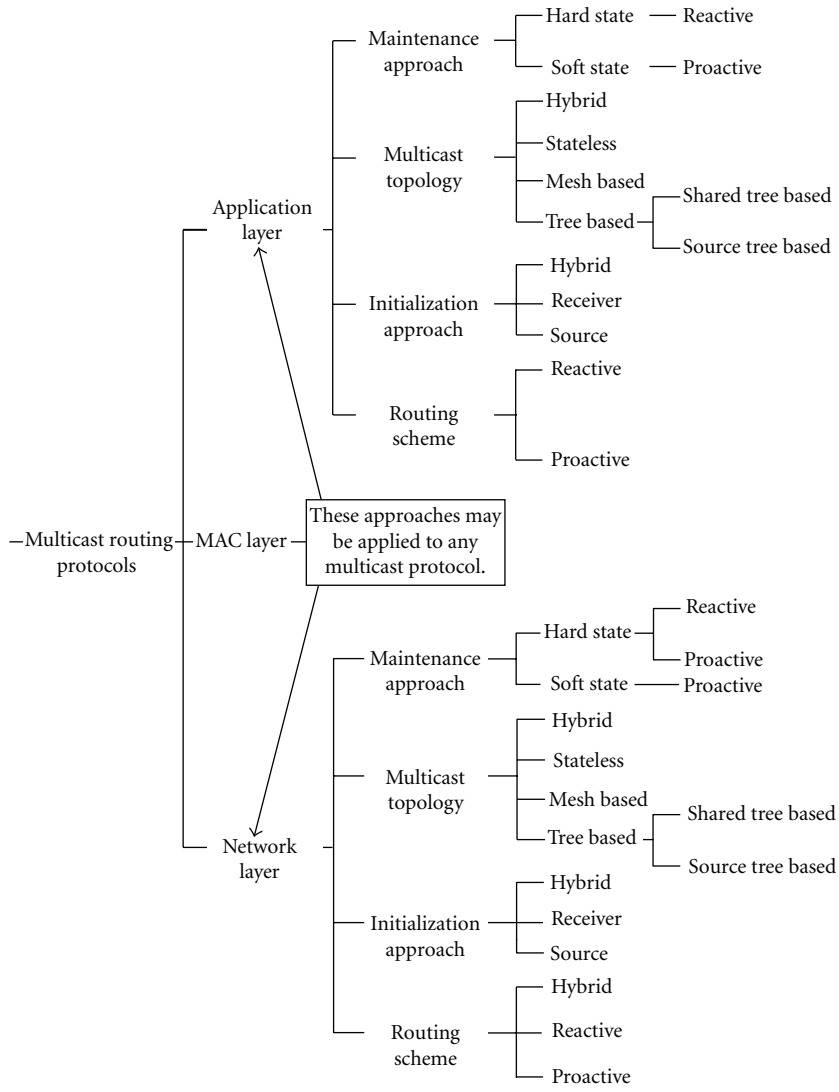


Figure 1: Classification of multicast routing protocols.

the network load is high and/or if there are a large number of multicast group members. In addition, since all multicast data packets are relayed from one group member to another in the form of a unicast packet, a large number of packet collisions and low resource utilization may result, especially where group member location density is high. Furthermore, the communicating member nodes are not aware of the increases in the physical hop count from the source node. As a result, in the case of mobility, using virtual links may lead to suboptimal paths (in terms of the number of hops). Reconfiguring the virtual connections is possible, but this introduces additional overhead. Overlay multicasting can improve routing efficiency by exploiting the broadcast nature of ad hoc networks. For instance, a one broadcast packet can be received simultaneously by two neighboring group members [18, 53].

MAC Layer Multicast (MACLM). Multicast data packets may need to be transmitted over many hops before the multicast reaches all its destination nodes. Since wireless links are prone to errors, multicast data packets may not always be received intact at the next hop along the path. Error recovery mechanisms may be deployed at the upper layer by requesting an *Ack* or feedback from the multicast destinations. This method requires nodes on the multicast tree (source node, destination nodes, and forwarder nodes) to buffer the multicast data packets until the feedback has been received. However, this method may cause significant end-to-end latencies in multicast data delivery, especially if the source and destination are separated by a large number of hops. In addition, this method may increase the node buffer size [54]. MAC layer multicasting is aimed at improving network efficiency through the implementation of positive *Ack* and retransmission policies for multicast data transmission. A reliable and efficient MAC layer multicast protocol can improve the performance of multicast communication. Table 1 presents a conceptual comparison of typical IPLM with ALM and MACLM.

Table 1: Conceptual comparison of IPLM, ALM, and MACLM.

Metrics	IPLM	ALM	MACLM
Multicast efficiency in terms of capacity/delay	high	low	high
Robustness	high	low	high
Control overhead	low	high	high
End-to-end delay	low	high	high
Ease of deployment	low	high	low
Packet delivery ratio	low	high	high

3.2. Table-Driven (Proactive) Approach versus Source-Initiated On-Demand (Reactive) Approach versus Hybrid Approach. Based on the routing information update mechanism (routing scheme) employed, multicast routing protocols for MANETs are classified into the following approaches.

Table-driven multicast routing protocols attempt to maintain consistent up-to-date multicast routing information between multicast group members in the network. These protocols require each node to maintain one or more table(s) to store routing information. In order to maintain a consistent network view, updates to the routing information tables are driven either by events (but only if a change is recognized) or periodically. As these protocols try to keep routing information up to date with topology changes, irrespective of whether or not this information is actually needed, they consume more power, and have high capacity and considerable control overhead, especially in a highly mobile environment where topology changes frequently. At the same time, these protocols have minimum route acquisition latency, since a route is always available to the source to reach a multicast group.

Source-Initiated On-Demand multicast routing protocols create routes only when desired by the source node (reactively). When the source node requires multicast routes to a multicast group, it initiates a route discovery process (local or global) within the network. Multicast routes and group membership are established, maintained, and updated on demand. Unlike *Table-driven* multicast protocols, *On-Demand* multicast protocols incur low control overhead, as well as saving on power and capacity. However, they may introduce route acquisition latency, since the source must wait until the multicast path has been discovered.

Hybrid multicast routing protocols, which attempt to combine the *Table-driven* and *Source-Initiated On-Demand* approaches at the same time, in order to alleviate the drawbacks of each. A proper proactive multicast routing approach and a proper reactive multicast routing approach are deployed at different hierarchical levels. Moreover, these protocols maintain the topology inside a zone with a certain radius (Intra-Zone) using the *Table-driven* approach, and outside this zone (Inter-Zone) using the *Source-Initiated On-Demand* approach. The main drawback of this approach is that a node outside the zone may wait a considerable time to join a multicast group.

3.3. Source-Initiated Approach versus Receiver-Initiated Approach versus Hybrid Approach. Based on how multicast connectivity is established and maintained, multicast routing protocols are classified into the following two approaches.

(a) The *Source-Initiated* approach, in which a multicast group is initiated and maintained by the source node (multicast group/source). The source constructs a multicast mesh or tree by flooding the network with a *Join Request* message. Any receiver node wishing to join a multicast group replies with a *Join Reply* message.

(b) The *Receiver-Initiated* approach, in which any receiver node wishing to join a multicast group floods the network with a *Join Request* message searching for a route to a multicast group. The management of the membership of a multicast group is usually assigned to a core (rendezvous) node. All sources of the same multicast group share a single multicast connection.

Some multicast protocols may not fall strictly into either of these two types of approach when they do not distinguish between source and receiver for initialization of the multicast group. Initialization is achieved either by the source or by the receiver. This type can be identified as a *hybrid* approach.

3.4. Tree-Based Approach versus Mesh-Based Approach versus Hybrid Approach versus Stateless Approach. Based on how routes are constructed for the members of the multicast group (network topology), multicast routing protocols for MANETs are classified according to the following types of approach.

Tree-based, in which a single path between source-destination pairs is established. There are two kinds of *Tree-based* approaches: *Source-Tree-based* and *Shared-Tree-based*. In the *Source-Tree-based* approach (persource tree), each source node creates a single multicast tree spanning all the members in a group. Usually, the path between the source and each member is not the shortest. In the *Shared-Tree-based* approach, only one multicast tree is created for a multicast group which includes all the source nodes. This tree is rooted at a node referred as the core node. Each source uses this tree to initiate a multicast.

Compared to the *Source-Tree-based* approach, the *Shared-Tree-based* approach is less efficient in multicast. In this one, the path between the source and the destination in the pair is not the shortest, but has a single point of failure and more overhead, since it maintains more routing information. In addition, the traffic is aggregated on the shared (backbone) tree rather than evenly distributed throughout the network, which gives it low throughput. Moreover, multicast protocols using a *Shared-Tree-based* approach require proper protocol operation to manage network partitions and mergers, since multicast group members may be separated into several disconnected partitions. However, multicast protocols intended for a *Source-Tree-based* approach do not require such protocol operations, because only the partition that includes the source maintains the multicast tree. In addition, the *Source-Tree-based* approach has a scalability

problem, but better throughput, since the traffic is evenly distributed throughout the networks.

In the *Mesh-based* approach, a multicast mesh connecting a source to all receivers in the network is constructed. There are multiple paths connecting the source and destination in the pair. These redundant paths provide more robustness (resilient to link failure) and higher packet delivery, but, at the same time, they introduce capacity wastage, power inefficacy, and more overhead because of data packet duplication. In contrast, the *Tree-based* approach is both capacity and power efficient, but more susceptible to link failure because of lack of node mobility. Finally, the *Mesh-based* approach is much more suitable than the *Tree-based* approach for MANETs.

In order to achieve both robustness and efficiency, the *Hybrid* approach attempts to combine both the *Mesh-based* and the *Tree-based* approaches.

Both these approaches have an overhead which is used to construct and maintain the delivery of the multicast tree or mesh, especially in an environment with frequent mobility. The *Stateless* approach is introduced to minimize the effect of this [23, 51, 55]. Instead of maintaining the routing information at every forwarding node, a source explicitly mentions the destination list in the packet header, and so this approach is intended for a small multicast group.

3.5. Soft-State Approach versus Hard-State Approach. MANETs suffer from frequent link breaks due to the lack of mobility of the nodes, which makes efficient group maintenance necessary. Maintaining the multicast group can be achieved by either the *Soft-State* approach or the *Hard-State* approach.

In the *Soft-State* approach, the multicast group membership and associated routes are refreshed periodically (*proactively*) by the flooding of control packets, whereas in the *Hard-State* approach, broken links are reconfigured by deploying two different approaches. The first is *reactive*, where routes are reconfigured, by sending control packets, only when a link breaks. The second is *proactive*, where routes are reconfigured before a link breaks, and this can be achieved by using local prediction techniques based on GPS or signal strength. The *proactive* approach is more reliable than the *reactive* approach, because it has much less packet loss, that is, it has a higher packet delivery ratio.

The Hard-State approach is much more efficient in terms of overhead. In contrast, the *Soft-State* approach is much more efficient in terms of reliability (packet delivery ratio). We can, therefore, conclude that there is a tradeoff between overhead and reliability.

4. Multicast Session Life Cycle

The various issues involved in a typical multicast session can be identified in the life cycle of the session. During that period, important events can occur: joining/leaving and rejoining a session, and session maintenance. These events can substantially affect the performance of multicast communication. Existing multicast protocols deploy different strategies to handle these events in order to maintain the quality of a multicast session (high packet delivery ratio, minimum end-to-end delay, etc.). This section describes how the session is established and terminated.

Before a source node sends multicast data, it checks whether or not the desired multicast group has been constructed. If it has, it sends multicast data immediately; otherwise, the source node must first construct it. Figure 2 describes a general method for initializing, constructing, maintaining, and terminating a multicast session.

When a source node has data to send, but no information on a route to a receiver is known, it floods a *Join Request* packet, as shown in Figure 2. Any node that receives a nonduplicate *Join Request* packet rebroadcasts the *Join Request* packet and stores the last hop node information in its routing table (i.e., a backward route). This process is continued until the *Join Request* packet reaches the receiver. The receiver replies with a *Join Reply* packet. When a node receives a *Join Reply* packet, it checks whether or not the next node address of the *Join Reply* entry matches its own address. If it matches, the node realizes that it is on the path to the source. Then, it marks itself as a *Forwarder Node* (node *J*, *K*, *X*, and *Y*). The *Join Reply* is propagated until it reaches the source node. This procedure constructs routes from the source node to all receivers. After these processes have been performed, the source can transmit multicast packets to receivers via selected routes and forwarder nodes. This method is known as *source-initiated*.

In a *receiver-initiated* method, if a node wants to join a multicast group (see the receiver at the bottom left of Figure 2), it broadcasts a *Join Request* packet. If the packet is received by a forwarder node, it replies with a *Join Reply* packet. If the *Join Request* packet is received by an intermediate node (nodes not on the tree, *A*, *B*–*E*, and *F*), it rebroadcasts the *Join Request* packet. This process is continued until it reaches a node on the tree (forwarder node or member node). The forwarder/member node replies with a *Join Reply* packet. When a node receives a *Join Reply* packet, it checks whether or not the next node address of the *Join Reply* entry matches its own address. If it does, the node realizes that it is on the path to the receiver. It then marks itself as a *Forwarder Node*. The *Join Reply* is propagated until it reaches the receiver node.

There are different mechanisms for maintaining the connectivity of the multicast group. First, the source (core node, group leader) of the multicast group periodically floods a control packet through the network during the *refresh period*; this is called the *Soft-State* method. The control packet is propagated by forwarder nodes, and it eventually reaches all the receivers of the multicast group. Any receiver who wants to leave the multicast group simply does not respond to the control packet; otherwise, it transmits a *Join Reply*. Second, the receiver node periodically floods a control packet through the network. Only source node or forwarder nodes are allowed to respond to the control packet; this is also known as a *Soft-State* method. Third, when a link break is detected between two nodes, a route repair procedure is carried out. One of these two nodes is responsible for detecting and repairing the broken link. This can be done

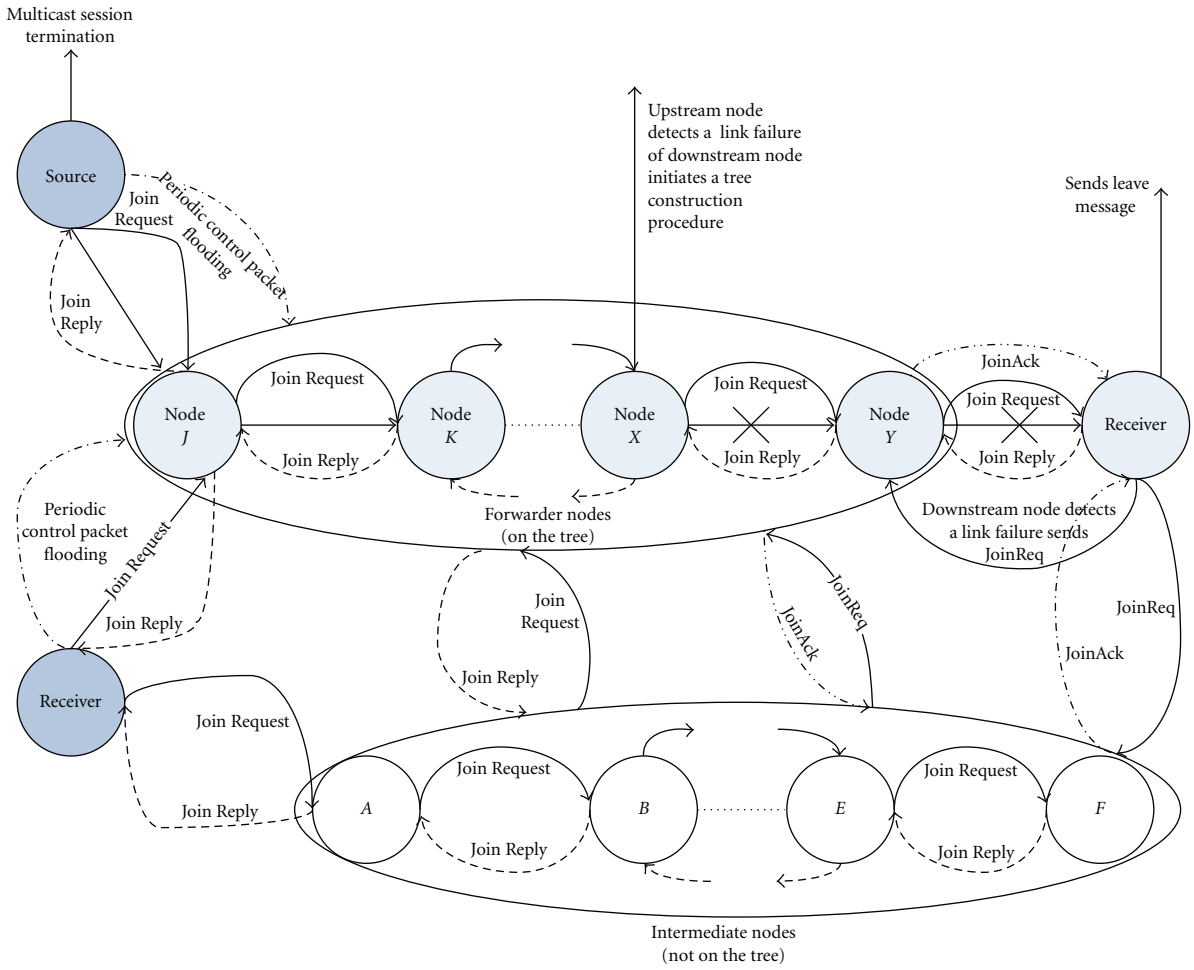


Figure 2: Multicast session life cycle.

in two ways. In the first, the downstream node (the furthest from the source/core/group leader node) sends a *Join Request* packet to search for its upstream node (receiver on the right-hand side of Figure 2) by limited flooding. If any node of the desired multicast group (forwarder node or group member) receives the *Join Request* packet, it replies with a *Join Acknowledgment* packet. Otherwise, the *Join Request* packet is rebroadcast by an intermediate node until it reaches a node of the desired multicast group. In the second, the upstream node (the nearest node from the source/core/group leader node) initiates a tree construction process (node *X* in Figure 2). The third mechanism for repairing a broken link is known as a *Hard-State* approach.

The multicast session is terminated by the source/core/group leader node by sending an *End Session* packet, or simply by stopping the transmission of multicast data. If a receiver node wants to leave a multicast group; it sends a *Leave* message or it does not respond to the *Join Request* message sent by the source during the *refresh period*.

5. Multicast Routing Protocols in MANETs

This section describes some of the existing multicast routing protocols used in MANETs. We classify them into three categories, according to their layers of operation. The categories are the network (IP) layer, the application layer, and the MAC layer.

5.1. Network Layer Multicasting (IPLM)

Associativity-Based Ad Hoc Multicast (ABAM)

Protocol Description. ABAM [19] is an *On-Demand Source-based* multicast routing protocol for mobile ad hoc networks. A multicast tree rooted at the multicast sender is established for each multicast session based primarily on association stability. Association stability helps the source to select routes to receivers which will probably last longer and need less reconfiguration. To initiate the multicast session, a multicast sender broadcasts a Multicast Broadcast Query (*MBQ*) message throughout the network. Nodes receiving the *MBQ* message will append their addresses and other information (route relaying load, associativity ticks, signal strength, power life) to the *MBQ* message before it is rebroadcast. Hence, each *MBQ* message accumulates information about the path traveled as it is forwarded. Multicast receivers will collect all the *MBQ* messages for the multicast group it wants to join. The most stable route back to the multicast sender

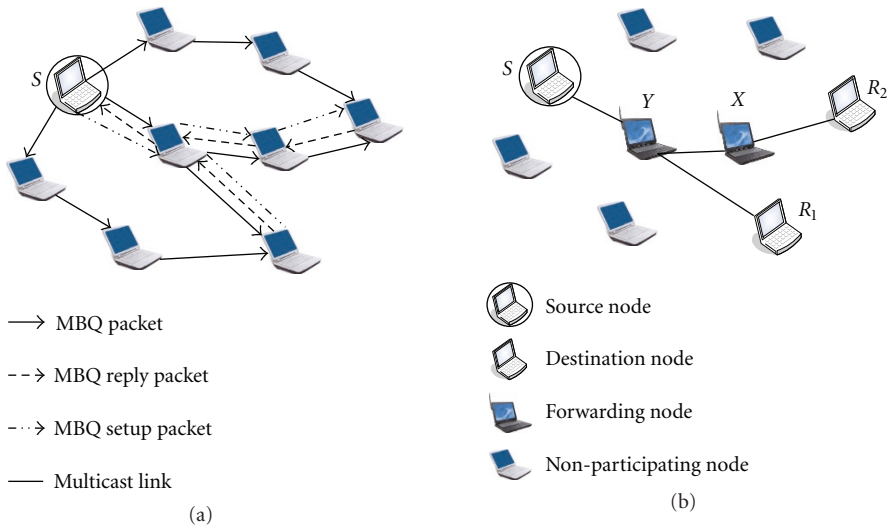


Figure 3: (a) Multicast tree construction in ABAM. (b) Multicast tree at the end of the construction.

will be chosen from all these possible routes and the *MBQ-Reply* message will be sent back to the multicast sender via the chosen path. Several *MBQ-Reply* messages, one from each multicast receiver, will be received by the multicast sender. With all received *MBQ-Reply* messages, the multicast sender will compute a stable multicast tree that results in shared links and generate an *MC-Setup* message to establish the multicast tree. That message will be propagated to all nodes along the tree, and these nodes will be programmed to participate in multicast forwarding. The tree construction phase is illustrated in Figure 3. A broken link is detected and repaired by the upstream node. When the upstream node, say node X, detects a broken link, it sends a *LocalQuery* packet (TTL = 1) searching its downstream node (receiver R_2). When the downstream node receives a *LocalQuery* packet, it replies with a *LocalQuery-Reply* packet and rejoins the multicast group. If the upstream node failed to find its downstream node, the next upstream node is responsible for repairing the broken link. This process terminates at a branch node Y (a node connecting many receivers). After that, R_2 sends a *JoinQuery* packet to join the multicast group. If a branch node Y moves, it sends a *LocalQuery* packet (TTL = 2, the number of hops to the furthest affected receiver on that broken branch) searching for the two receivers R_1 and R_2. When a receiver leaves a multicast group, it sends a *Leave* message, which results in the branch being pruned (if there are no other receivers in that branch). When a multicast group has no more receivers, that is, when all the members have decided to leave the group, the tree will be pruned incrementally. The multicast tree can also be deleted when the source no longer wishes to act as a multicast sender. It can do this by sending a multicast *Delete* message to prune the tree.

Discussion. ABAM introduces less control overhead traffic and achieves a higher packet delivery ratio in comparison with ODMRP [35], due to the stability of the path between the source and destination nodes. At the same time, the path may be long, and some latency in delivering the data packets will be incurred. In addition, ABAM suffers from scalability issues.

Differential Destination Multicast (DDM)

Protocol Description. DDM [23] is a receiver-initiated multicast routing protocol. It operates in two modes: Soft-State and Stateless. In Stateless mode, source nodes insert the destination address into the field, called the DDM block of the data packet, and unicast it to the next node, using the underlying unicast routing protocol. Every such node that receives the DDM block data packet acquires the address of the following node and unicasts the DDM block data packet. Finally, the data packet reaches its destinations. In this way, the protocol avoids maintaining multicast states in the nodes. The tree initialization phase is illustrated in Figure 4. In soft-state mode, each node along the forwarding path remembers the destination address by sorting it in the forwarding set. Therefore, by caching this information, there is no need to list all the destination addresses in every packet, which is why it is called the Differential Destination Multicast protocol. This protocol is best suited for applications with small multicast groups in a dynamic MANET environment.

Discussion. DDM consumes a significant bandwidth, since each destination periodically sends *Join* control packets to the source to show its interest in the multicast session. In addition, the size of the DDM block data packet becomes larger as the number of receivers increases, which means that it is not scalable. DDM operates in centralized fashion (the source node manages group membership), and, therefore, security is ensured. Finally, DDM requires minimum memory resources, since it operates in a Stateless fashion.

Bandwidth-Efficient Multicast Routing (BEMRP)

Protocol Description. BEMRP [20] is aimed at designing a multicast routing protocol that uses bandwidth efficiently by

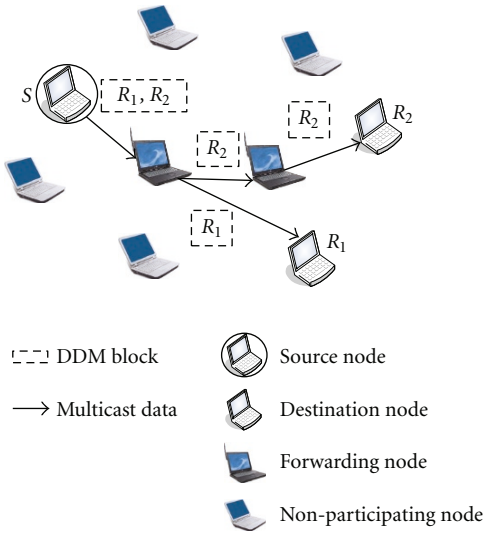


Figure 4: Multicast tree in DDM.

constructing a receiver-initiated tree-based multicast source. It finds the nearest forwarding group member nodes for broadcasting *Join* requests, instead of finding the shortest path from source to receiver, thereby reducing the number of data packet transmissions. All nodes on this path then become forwarding nodes. The unwanted forwarding nodes are removed using route optimization, which reduces the number of data packet transmissions and saves bandwidth. When a receiver node X wants to join a multicast group, it broadcasts a *Join* packet. *Join* packets are flooded until they reach a forwarding node or a receiver node of the multicast group. A forwarding node or a receiver node waits until they receive a certain number of *Join* packets or reach some predetermined time, and then choose a *Join* packet with the smallest hop count. *Reply* packets are sent back to node X, following the reverse path that the selected *Join* packet has traveled. Node X also waits until it receives a certain number of *Reply* packets or reaches some predetermined time, and then chooses a *Reply* packet with the smallest hop count. Figure 5 illustrates the joining process. When a node X, which is a receiver node of a multicast group, wants to leave the multicast group, it sends a *Quit* packet to its upstream node. Upon receiving the *Quit* packet, the upstream node simply deletes node X from the downstream entry in a multicast routing table, provided it has no other downstream nodes. Otherwise, it sends a *Quit* packet to its upstream node and leaves the multicast group. BEMRP follows the *Hard-State* approach to maintain the topology. Moreover, to rejoin the multicast group, a node transmits the required control packet after the link breaks.

Discussion. BEMRP follows the traditional multicast approaches, that is, distributed multicast routing state maintenance and distributed group membership manage-ment; hence, it suffers from security and resource use issues. BEMRP introduces some delay into delivering the multicast packets, since the paths between the source and the receivers are not optimal, and since a node spends some time repairing broken links and then rejoins the multicast group, creating even more delay in packet delivery. In addition, the distance between source and receiver is increased, which leads to an increase in the probability of path breaks; hence, the packet delivery ratio is reduced. Instead of using the shortest source-receiver pair path, it tries to find the nearest forwarding node, thereby reducing the number of data packet transmissions, which results in a saving of bandwidth. The proactive *Hard-State* approach also helps BEMRP to save bandwidth by only transmitting the control packets after the link failure, although this may introduce some latency.

Weight-Based Multicast Protocol (WBM)

Protocol Description. WBM [43] is a receiver-initiated multicast routing protocol. It uses the concept of weight when deciding upon the entry point in the multicast tree where a new multicast member node is to join. Moreover, when a new receiver X decides to join the group, it broadcasts a *JoinReq* packet with a certain time-to-live (TTL) entry. These *JoinReq* packets are forwarded until they are received by a tree node. Upon receiving a *JoinReq* packet, a tree node, say node W, sends a *Reply* packet. Several such replier nodes can send *Reply* packets, which initially contain the hop distance of the node W from the source S, and also the hop distance of the node X from node W. As *Reply* packets are forwarded, the hop count taken from the replying node W is maintained in the *Reply* packet. Thus, the *Reply* packet, when it arrives at a receiver node X, will have the hop distance of the node X from node W and the hop distance of node W from the source S. The joining process is illustrated in Figure 6. If node X joins the multicast group through node Z, then the hop distance of the destination X from the source node S will only be 3 at the cost of two additional forwarding nodes. If it joins through node Y, then no additional forwarding node need to be added. This is at the cost of increased hop distance, which is 6 in this case. A parameter called *joinWeight* (which governs the behavior of the protocol) tries to find the best path by considering not only the number of added forwarding nodes but also the hop distance between the source and destination. After receiving a number of *Reply* packets, the node maintains a best *Reply*, which is updated when new replies are received. The best *Reply* minimizes the quantity, $Q = (1\text{-}joinWeight) * (hop\ distance\ of\ X\ from W - 1) + joinWeight * (hop\ distance\ of\ X\ from\ W + hop\ distance\ of\ W\ from\ S)$. A timer is set upon receipt of the first *Reply* packet. Once the timer expires, node X sends a *JoinConf* message along the reverse path that the selected *Reply* has traveled.

Discussion. The weight concept provides flexibility for a receiver to join either the nearest node in the multicast tree or the node nearest to the multicast source, resulting in high efficiency of the protocol. Due to the dependence of the weight on several factors, such as the size of the multicast group and the network load, it is considered to be a disadvantage. WBM uses a localized predication technique that avoids path breaks. Packet loss is, therefore, low, resulting in a high packet delivery ratio. However, the

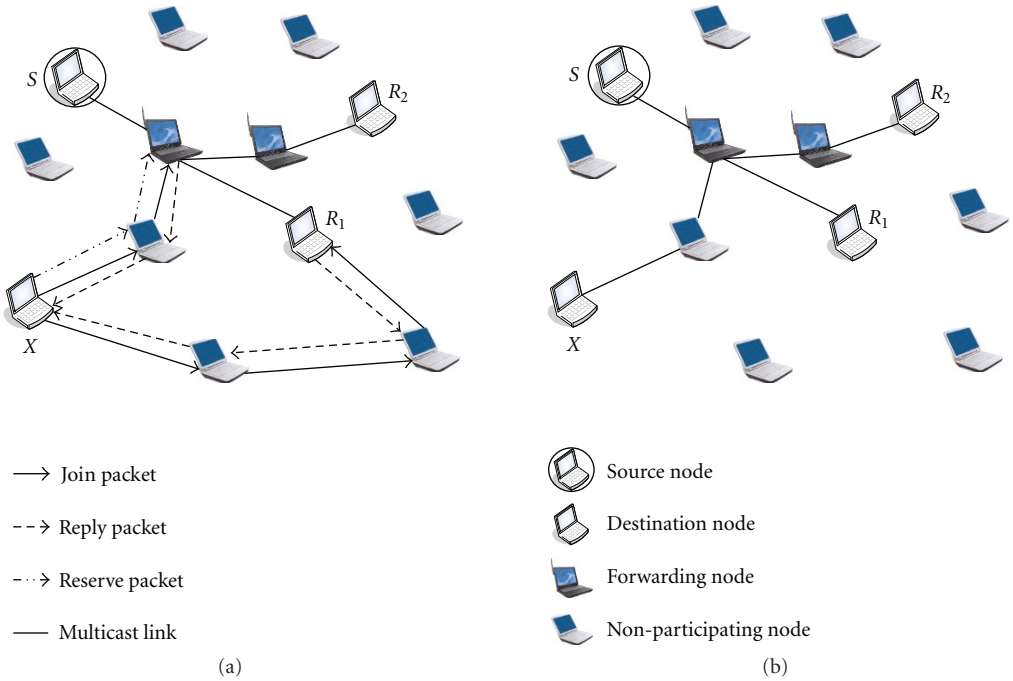


Figure 5: (a) Node X joins a Multicast group in BEMRP. (b) Multicast tree at the end of the joining process.

predication technique may not work well, for example, in a high-fade environment.

Multicast Routing Protocol Based on Zone Routing (MZRP)

Protocol Description. MZRP [32] is a source-initiated multicast protocol that combines reactive and proactive routing approaches. Every node has a routing zone. A proactive approach is used inside this zone and a reactive approach is used across zones. First, a source node constructs a multicast tree inside its routing zone, and then it tries to extend the tree outside the zone (the entire network). A node (which is already a multicast forwarding node for that group), wishing to join a multicast group, changes its status from multicast forwarding node to multicast group member. Any other node sends a multicast route request (*MRREQ*) message. There are two kinds of *MRREQ*, unicast or broadcast, depending on the information the source node has. If the source node has a valid route to any node on the tree and it wants to join that group, it sends a unicast *MRREQ* along the route to the multicast tree and waits for a multicast route reply, *MRREP*. The intermediate nodes forward the unicast *MRREQ* and reverse paths are set in their multicast routing tables. When the destination receives the *MRREQ*, it sends an *MRREP*. If the unicast *MRREQ* fails or the source node does not have a valid route to that group, it initiates a bordercast *MRREQ*, which is sent via the bordercast tree of the source node. When the bordercast *MRREQ* reaches the peripheral nodes, they will check whether or not they have a valid route to that multicast group or group leader. If so, they will send unicast *MRREQs* instead of bordercast *MRREQs* and wait for the *MRREPs*. Otherwise, bordercast *MRREQs* will be sent via the bordercast tree of the peripheral nodes, and so forth. Reverse paths will be established among the intermediate nodes. When a destination node receives an *MRREQ* for a multicast group, and if it is a multicast tree member of that multicast group, it will send an *MRREP* to the source and wait for the multicast route activation *MRACT* message from the source node to activate the new branch of the multicast tree. The *MRREP* is sent to the source along the reverse path. Figure 7 shows the construction of a multicast tree in MZRP. A multicast group member wanting to leave the group will, if it is a leaf node on the multicast tree, prune itself from the tree by sending a multicast prune message *MPRUNE* toward an upstream node. The upstream node also will prune itself from the tree if it is not a group member, and becomes a leaf node. Otherwise, the pruning procedure will stop.

Discussion. MZRP scales well for different group sizes. MZRP runs over the Zone Routing Protocol (ZRP) [56], so the two can exchange information, which means that MZRP has less control overhead than ODMRP. One of the main drawbacks of this protocol is that a node outside a source routing zone will wait a considerable time to join the group. Compared with the Shared-Tree-based approach, MZRP creates many more states at nodes involved in many groups, each with multiple sources.

Multicast Core Extraction Distributed Ad Hoc Routing (MCEDAR)

Protocol Description. MCEDAR [28] is a Source-Tree-based multicast protocol. It combines the Tree-based protocol and the Mesh-based protocol to provide efficiency. It uses CEDAR [57] to construct the mesh. MCEDAR uses

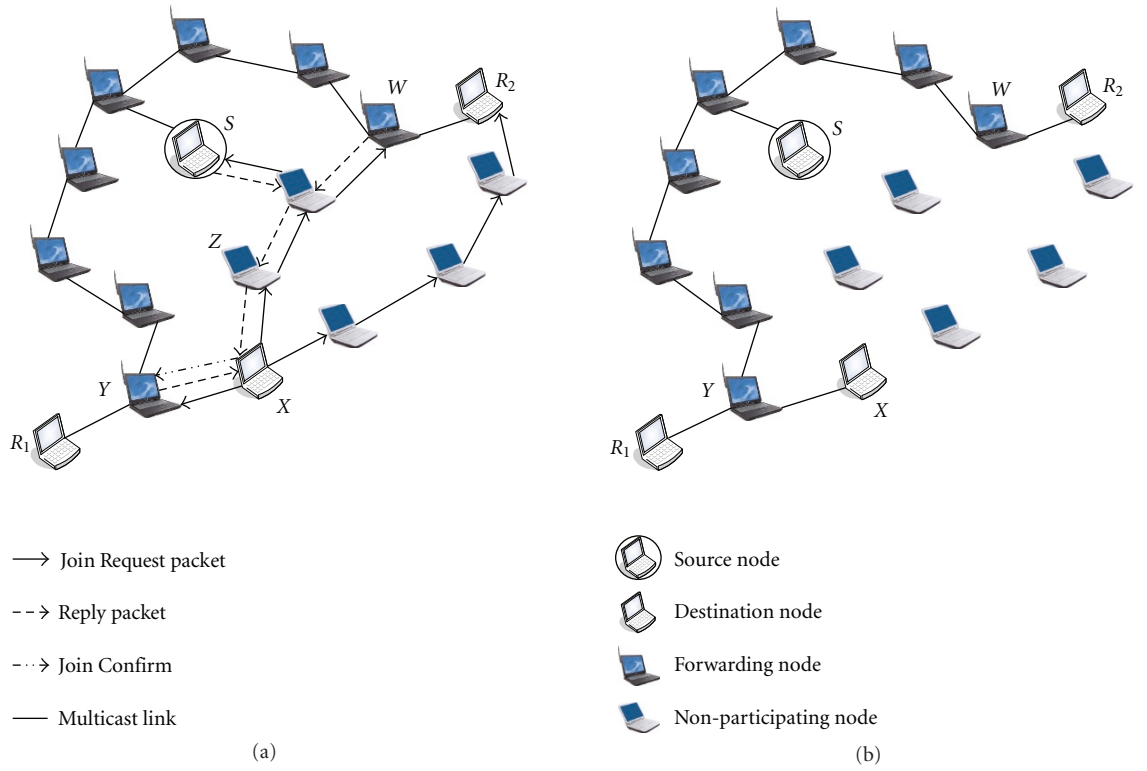


FIGURE 6: (a) Node X joins a Multicast group in WBM. (b) Multicast tree at the end of joining process.

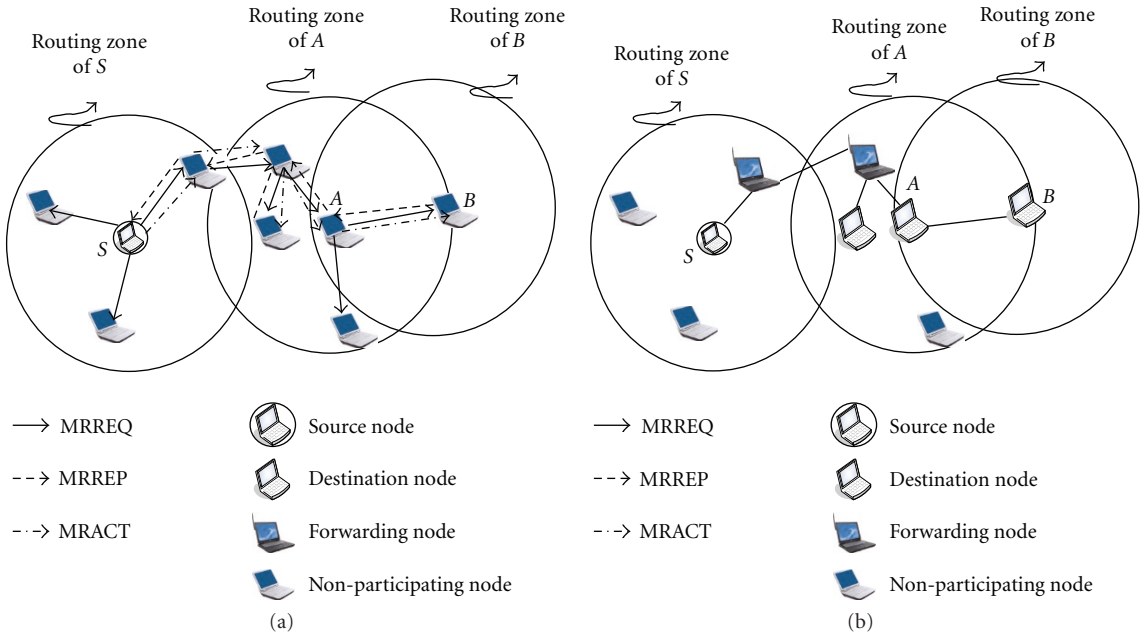


FIGURE 7: Multicast initialization in MZRP: (a) the source node sends MRREQ, (b) tree construction.

a mesh structure called the *mgraph* as its multicast routing infrastructure. CEDAR creates a minimum dominating set (MDS) of core nodes using a core computation algorithm. In addition, CEDAR provides a mechanism for core broadcast on reliable unicast, which dynamically establishes a source tree. Each core in this set advertises its existence through a beacon signal up to the next 3 hops, and, therefore, each core identifies its nearby cores and builds a virtual link. Every nonmember node located 1 hop away from at least one core node selects one of the core nodes as its dominator

node. When a noncore node wants receiver R_1 to become a member of a multicast group, it requests its dominating core node, core 5, to perform the join operation. A node performs the join operation by core broadcasting a *JoinReq (MA, joinID)*, which consists of the address of the group (MA) the node wishes to join and the current *joinID* of the node corresponding to the multicast group. When a node that is not a member of the multicast group (MA) receives the *JoinReq*, it forwards the message to its nearby core nodes in accordance with the core broadcast mechanism. In contrast, when an existing MA member receives the *JoinReq*, it sends a *Join-Ack (MA, joinID)* only if its *joinID* is smaller than the *joinID* that arrives in the request. It then forwards the *JoinReq* further downstream. However, if its *joinID* is larger than the incoming *joinID*, it forwards the request like a nonmember. The *joinID* in the *Join-Ack* message sent back to the node requesting the join is that of the replying node. When an intermediate node on the reverse path receives the *Join-Ack* message, it decides whether to accept it or reject it based on the robustness factor (R). Each *mgraph* member maintains two other data structures, the *parent* set and the *child* set. When a node accepts a *Join-Ack*, it adds the upstream *mgraph* member to its parent set. Further, if the downstream node is not already in its *child* set, it forwards the *Join-Ack* to the downstream node and adds the downstream node to its *child* set. However, when the intermediate node decides to reject a *Join-Ack*, it suppresses the *Join-Ack* and performs an explicit leave from the upstream node so that its ID is removed from the upstream node's *child* set. The number of accepting *Join-Ack* packets at the dominator node (core 5) is governed by the robustness factor (R). If $R = 2$, therefore, core 5 will accept only two *Join-Acks* and reject the others. The member on accepting a *Join-Ack* sets its *joinID* to the maximum of its current *joinID* and the arriving *joinID* incremented by one. It then stamps the *joinID* of the *Join-Ack* with its new *joinID*. Figure 8 illustrates the joining process of the new receiver R_1 with *joinID* = 6. Figure 9 shows how data are forwarded in MCEDAR.

Discussion. MCEDAR is robust and efficient, since a receiver node has multiple paths to a multicast tree. However, when used with small and sparsely distributed groups, it may become less efficient and more expensive due to bandwidth constraints, network topology dynamics, and high channel access cost. In a high mobility environment, nodes need to change their cores frequently, thereby increasing control overhead. MCEDAR is also more complex than other multicast routing protocols (Tree-based and Mesh-based).

Independent Tree Ad Hoc Multicast Routing (ITAMAR)

Protocol Description. ITAMAR [27] provides several heuristic schemes for constructing multiple independent trees. The multiple backup-independent trees are computed with minimal overlap, such that a tree is used until it fails and then is replaced by an alternative tree. Independent trees are computed by minimizing the number of edges and nodes that are common to the trees, under the assumption that node movements are independent of one another. This protocol is aimed at improving the average time between multicast tree failures. Moreover, new trees are computed when the probability of failure for the current set of trees rises above a threshold. In the case of mobility, it is important to estimate the time this happens, then, instead of replacing a tree if even one link fails, an independent path algorithm can find a set of backup paths to replace the damaged part of the tree.

Discussion. ITAMAR allows some overlapping, since totally independent trees might be less efficient and contain more links. As a result, the correlation between the failure times of the trees is minimal, which leads to improved mean times between route discoveries. At the same time, this will lead to a computationally intensive operation and may not be convenient in all situations. ITAMAR is basically based on the Dijkstra Shortest Path First (SPF) algorithm, and, therefore, needs to know the network topology in advance in order to construct multiple edge disjoint or nearly disjoint multicast trees in a centralized way. Therefore, it has a scalability issue, and also significant overhead will be incurred.

Preferred Link-Based Multicast (PLBM) Protocol

Protocol Description. PLBM [39] is a tree-based receiver-initiated protocol. It is an extension of the Preferred Link-Based Routing Protocol (PLBR) [58]. It uses only a set of links to neighboring nodes for forwarding *Join Query* packets (preferred links). Each node maintains two tables, a *Neighbor Neighbor Table* (NNT, for local network topology information) and a *Connect Table* (CT, for multicast tree information). Every node in the network periodically sends small control packets, called *beacons*. On receiving a *beacon*, a node updates the corresponding entry in its NNT. Thus, the NNT is kept up to date by means of the *beacon* packets. When a new member wishes to join the multicast group, it first checks its NNT to determine whether or not there are tree nodes (members, forward nodes, or multicast sources) in its NNT. If so, it sends a *Join Confirm* message to one of them without flooding the networks with any *Join Query* packet. Otherwise, it propagates a *Join Query* message if at least one eligible neighbor node is present in its NNT for further forwarding of the *Join Query* packet. The eligibility of a neighbor node to further forward the *Join Query* packet is determined using PLBR [58]. Only preferred nodes are eligible for further processing of the *Join Query* received. On receiving this packet, a node first checks its eligibility to forward it. If it is not eligible to do so, the packet is discarded. If an eligible node is connected to a multicast tree, it sends a *Join Reply* packet back to the node that originated the *Join Query* packet and starts a timer waiting for a *Join Confirm* packet from the node. Otherwise, it forwards the *Join Query* packet. The *Join Reply* packet follows the route traveled by the *Join Query* packet, but in the reverse direction. Figure 10 shows multicast tree initialization and construction phases in PLBM. In Figure 10(a), a destination node R_2 sends *Join Query* packet to nodes A and B based on the *Preferred* List (PL, a subset of nodes which are selected by a node from its neighbor list (NL) based on node or link characteristics)

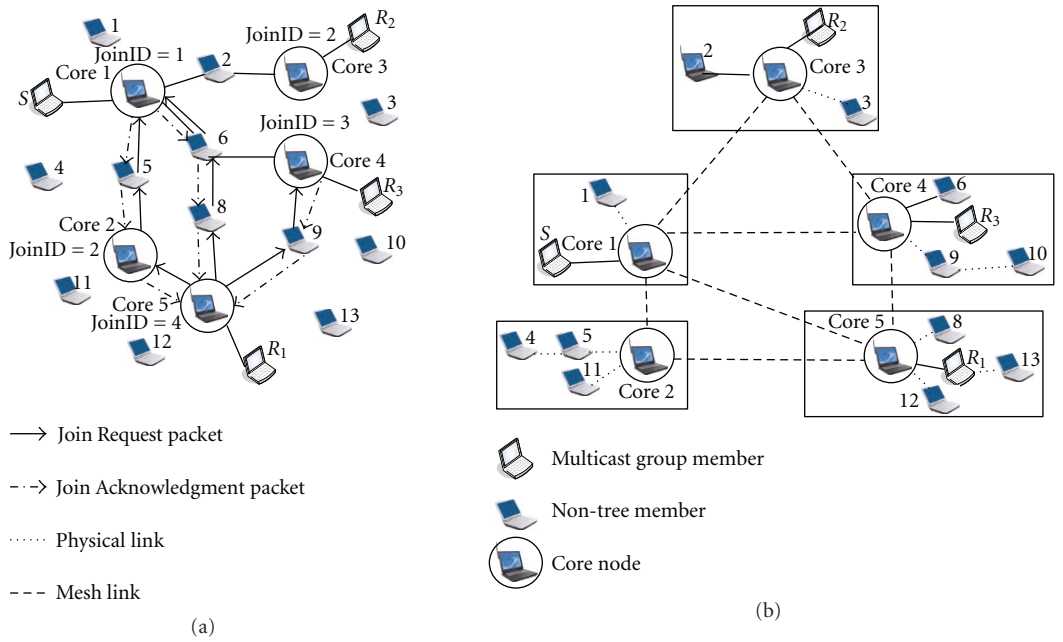


Figure 8: (a) Core 5 sends a JoinReq packet in MCEDAR. (b) Virtual multicast mesh.

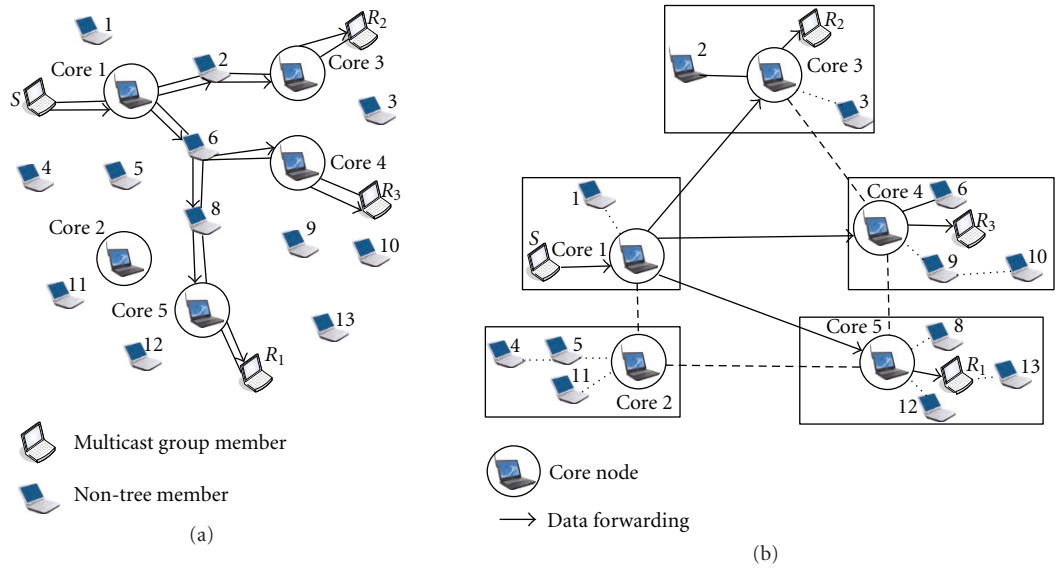


Figure 9: Multicast data forwarding.

using PLBA. When nodes A and B receive a *Join Query* packet, they also compute their preferred neighbors using PLBA. Therefore, nodes A and B send *Join Query* packet to $\{C, D\}$ and $\{E, F, G\}$, respectively. Nodes E and D drop the *Join Query* packet, because they do not have any preferred nodes. Nodes C, F, and G forward the *Join Query* packet to nodes K, K, and G, respectively. Eventually, the source node S receives a single *Join Query* packet. After that, the source node S sends the *Join Reply* packet through path 1 ($S \rightarrow K \rightarrow G \rightarrow B \rightarrow R_2$) and path 2 ($S \rightarrow K \rightarrow F \rightarrow B \rightarrow R_2$). Finally, the destination node R_2 selects the first *Join Reply* packet it receives and sends *Join Confirm*, assuming that path 1 is selected. If a node, say node R_3, wants to join the multicast group and it has a tree node (member nodes or forwarding nodes) in its NNT, it sends a *Join Confirm* packet to the tree node, say node B (forwarding node), without flooding the network with the *Join Query* packet. However, if a node wants to join a multicast group but does not have a tree node in its NNT, it (say node R_1) propagates a *Join Query* packet to be flooded in a limited manner through the network based on PLBA. Nodes D and M receive the *Join Query* packet. After that, nodes D and M send the *Join Query*

packet directly to their tree node, nodes B and G, respectively. Finally, node R_1 receives two *Join Reply* packets (from nodes B and G). R_1 then selects the first *Join Reply* packet it receives (assuming it selects path $S \rightarrow K \rightarrow G \rightarrow C \rightarrow M \rightarrow R_1$), sends a *Join Confirm* packet, and joins the multicast group.

Discussion. It has been reported in [45] that the concept of the preferred link involved in PLBM provides better adaptability and flexibility. In addition, the use of 2-hop local topology information provides efficient multicast routing. The preferred list may be based on other link or node characteristics, for example, delay, bandwidth, and stability, which enables the PLBM protocol to take into consideration the QoS requirements. Since every node in the network sends a *beacon* packet periodically, considerable control overhead is introduced.

Probabilistic Predictive Multicast Algorithm (PPMA)

Protocol Description. PPMA [44] tracks relative node movements and statistically estimates their relative positions in the future to maximize the multicast tree lifetime by exploiting more stable links. In order to remedy drawbacks of this protocol, which are lack of tree robustness and lack of reliability in highly mobile environments, PPMA continuously tracks the evolution of the network state; it defines a probabilistic link cost as a function of energy, distance, and node lifetime; and it tries to keep all the nodes alive as long as possible. Also, PPMA takes into account the estimated network state evolution in terms of residual node energy (low-energy nodes cannot join multicast trees), link availability, and node mobility forecast, in order to maximize the multicast tree lifetime. The PPMA algorithm has a centralized and a distributed version. In the centralized version, a node has a set of potential fathers for a given number of hops. Higher priority is given to those nodes within the transmission range that have other children, in order to exploit the broadcast property of the wireless medium. The closest of the potential fathers is chosen for power efficiency reasons. In the distributed version, a private cost is defined to find the minimum cost path to the source, in addition to a public cost to enable a node to join a tree. A new receiver finds the best public cost path and joins the tree, whereas an old receiver changes its path if it finds a lower private cost. The cost can typically be an entity, such as energy consumption. The closest of the potential fathers is chosen for power efficiency reasons.

Discussion. PPMA overcomes the tradeoff that exists between the bandwidth efficiency to set up a multicast tree and the robustness of the tree based on node energy consumption and mobility, by decoupling tree efficiency from mobility robustness. PPMA exploits the nondeterministic nature of ad hoc networks by taking into account the estimated network state evolution in terms of residual node energy, link availability, and the node mobility forecast, in order to maximize the multicast tree lifetime. However, the path between nodes is not the shortest, and so a significant control overhead will be incurred to maintain the path at different nodes and the end-to-end delay will also be increased.

Adaptive Demand Driven Multicast Routing (ADMR) Protocol

Protocol Description. ADMR [15] maintains a tree for every source-multicast pair. Each tree is maintained by a periodic flood of *keep alive* packets within the tree. The Multicast Routing state in ADMR is dynamically established and maintained only for active groups with at least one receiver and one active sender in the network. Each multicast data packet is forwarded from the sender to the receivers along the shortest delay path with the multicast forwarding state. Senders are not required to start or stop sending data to the group, or to join the group to which they wish to send. Furthermore, receivers dynamically adapt to the sending pattern of senders and mobility in the network. ADMR also detects when mobility in the network is too high to efficiently maintain the multicast routing state, and instead reverts to flooding for a short period of time if it determines that the high mobility has subsided. ADMR monitors the traffic pattern of the multicast source application, and, based on that, can detect link breaks in the tree, as well as sources that have become inactive and are no longer sending any data. In the former case, the protocol initiates local repair procedures and global repair if the local repair fails. A multicast state setup starts when a new multicast source node S starts sending to a multicast group G for which at least one receiver exists in the network, or when a receiver joins a multicast group G for which there is at least one source in the network. The source node S sends a multicast packet targeted at group G when no routing state yet exists for this source and group. The routing layer on S adds an ADMR header to the data packet and sends the data packet as a network flood. Each node in the network that receives this packet forwards it unless it has already forwarded a copy of it. In addition, the node records the MAC address of the node from which it received the packet in its *Node Table*, and the *sequence number* stored in the packet's ADMR header. This information will not only be used for duplicate detection but also for forwarding packets back to S. Furthermore, receivers for group G send a *Receiver Join* packet back toward S. Every node that forwards this packet creates a forwarding entry in its *Membership Table* for source S and group G, indicating that it is a forwarder for this sender and this group. The collection of paths with forwarding state between S and the receivers for G produces the Forwarding Tree. Figure 11 illustrates the multicast state setup.

Discussion. ADMR adapts well to the network load, and also avoids unnecessary redundancy. One of its shortcomings is that a large amount of state information needs to be maintained at every node for every group source. Joining a group is very costly. A receiver must first send a flood, and then each source must reply to the new receiver. The receiver must then send a confirmation to every source. This is especially costly if the tree breaks often and the receiver is repeatedly trying to join the group. Finally, the protocol

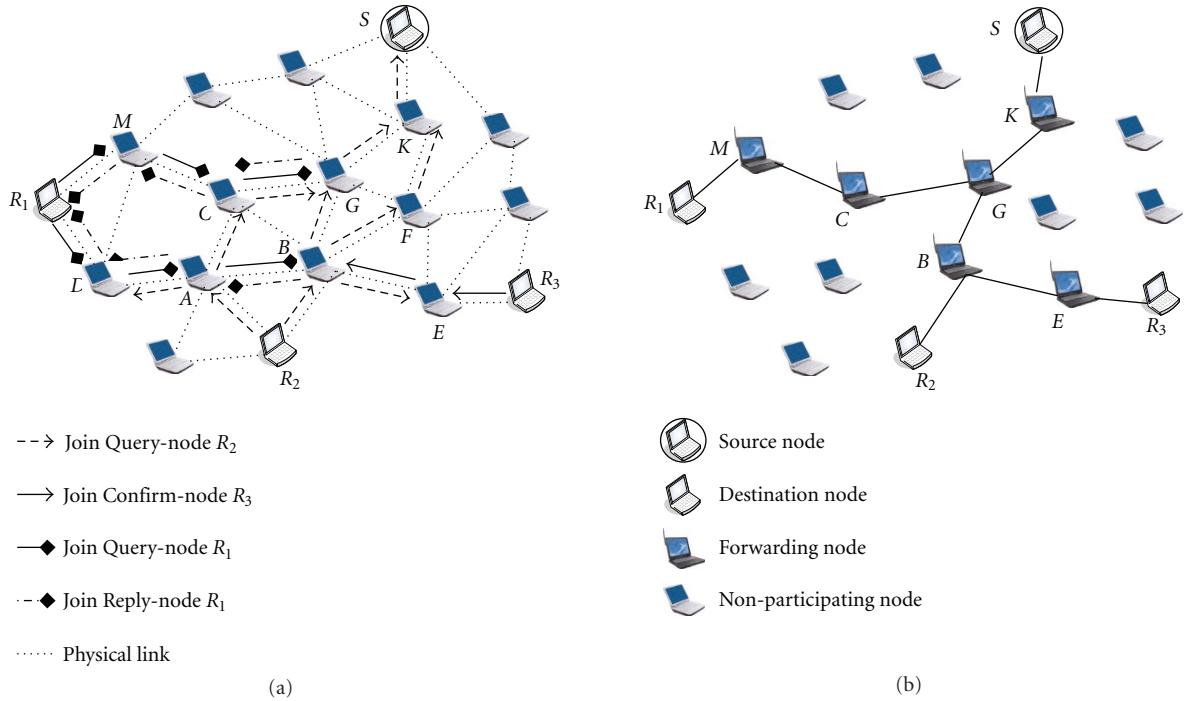


FIGURE 10: Multicast tree in PLBM: (a) initialization phase, (b) tree construction phase. © IEEE 2003.

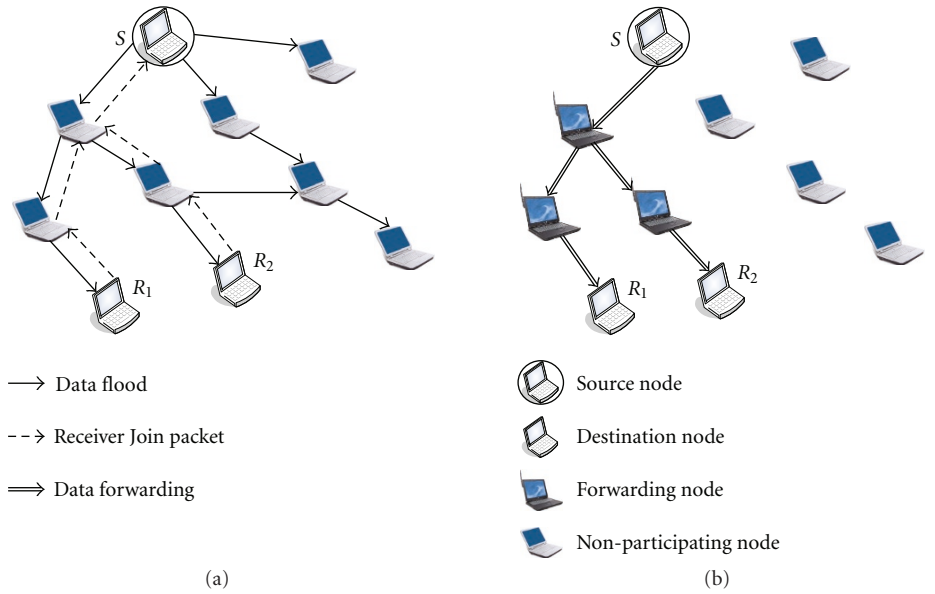


FIGURE 11: (a) Multicast tree construction in ADMR. (b) Multicast data forwarding.

indicates how the source moves to flooding mode for high mobility, but does not indicate how it moves back to a lower mode when mobility is reduced.

Multicast Ad Hoc On-Demand Distance Vector (MAODV)

Protocol Description. The MAODV [29] protocol is extended from AODV [59]. It maintains a shared tree for each multicast group, which consists only of receivers and relays (forwarding nodes). It determines a multicast route on demand by using a broadcast route discovery mechanism. The first member of a multicast group becomes the leader of that group. The multicast group leader is responsible for maintaining the multicast group sequence number and broadcasting this number to the multicast group. This is done through a group *HELLO* message. Nodes use the group *HELLO* information to update their *Request Table*. In Figure 12, if node R_3 wants to join a multicast group, it

originates a route request (*RREQ*) packet and unicasts it if it has the address of the group leader. If the address of the group leader is unknown, then R_3 broadcasts the *RREQ* packet, as depicted in Figure 12(a). Only the group leader, or a member of the desired multicast group with a sequence number larger than that in the *RREQ* packet, can respond to a *Join RREQ* packet. When the group leader or a member of the desired multicast group receives multiple *RREQ* packets, it selects the one with the highest sequence number and the lowest hop count, and unicasts a route reply *RREP* packet to the requesting node (the group leader and the forwarding node *X* unicast *RREP* packet in Figure 12(a)). The *RREP* packet contains the distance of the replying node from the group leader and the current sequence number of the multicast group. When the receiving node receives more than one *RREP* packet, it selects the most recent one and the shortest path from all the *RREP* packets. Then, it sends a multicast activation message *MACT* to its next hop to enable that route. Figure 12(b) shows the multicast tree at the end of the joining process. If a nonleaf node wishes to leave a multicast group, it sends a multicast activation message to their next hop with its prune flag set and prunes itself; otherwise, it cannot leave and must remain on the tree. MAODV employs an expanding ring search (ERS) to maintain the multicast tree. When a broken link is detected between two nodes, the downstream node is responsible for initiating the repair link. The downstream node broadcasts an *RREQ* packet using an ERS. Only the node with a hop count to the multicast group leader less than or equal to the indicated value in the *RREQ* packet can respond. If the downstream node does not receive a reply, it realizes that the multicast tree is partitioned. The downstream node becomes the new multicast group leader for its participation in the multicast tree partition. The multicast tree remains partitioned until the two parts of the network become connected once again.

Discussion. 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.

Mobile Multicast Agent (MMA)

Protocol Description. MMA [30] uses mobile multicast agents (MMAs) to form the virtual backbone of an ad hoc network. The MMA multicast algorithm is based on AODV [59] and provides multicast tree discovery and multicast tree maintenance. Moreover, it has a two-level hierarchy, where a special subset of network nodes forms a spine to act as a virtual backbone on top of a clustered structure. Spine nodes are known as MMAs, and are responsible for multicast tree discovery and maintenance. MMAs are also used as relay nodes, so that the multicast tree is composed of a sender node, MMAs, and multicast group members. When a mobile node wants to send a packet to a multicast group, it sends an *RREQ* packet to its MMA. If there is valid information for routing to the multicast group stored in the MMA, the MMA will reply with an *RREP* packet. If not, the sender should initiate a route request process. To limit the number of *RREQ* packets propagated, an MMA processes an *RREQ* packet only if it has not already seen the packet. In symmetric link ad hoc networks, an intermediate MMA can deliver the *RREP* packet on the reverse route of the *RREQ* packet, while in asymmetric link ad hoc networks, an intermediate MMA must initiate a test route discovery to the MMA of the sender node and piggyback the *RREP* packet on this new route request. Once the multicast routing tree discovery procedure is completed, data packets can be easily delivered to next hop from the sender along the multicast routing tree. Figure 13 shows a spine-based ad hoc network with 4 clusters. Multicast routing tree maintenance is based on the mobility information of both MMAs and nonspine nodes. A route error (*RERR*) packet and *ACKs* are used for route maintenance.

Discussion. According to this algorithm, only MMAs are used to transfer control information and retransmission packets, which means that control overhead and battery power are reduced and the throughput of the network is increased. Route information is only stored in MMAs, which reduces the time it takes to find the multicast tree and the time required for a sender node to obtain routing information. As the fulfillment of many responsibilities relies on MMAs, these nodes must have large buffer memories compared to other nodes.

Adaptive Shared-Tree Multicast (ASTM) Routing

Protocol Description. ASTM [9] is a hybrid protocol that combines the advantages of persource and shared trees and is based on the notation of the Rendezvous Point (RP). The RP-rooted multicast forwarding tree is created by receiver members periodically sending *Join Requests* to the RP. The *Join Request* contains the forward list, which is initially set to include all senders. Sources send their multicast data to the RP, and the RP forwards the multicast data to the receivers. Internal nodes on the path between the source and the RP may not forward these packets to other nodes if the protocol is operating in the unicast sender mode. However, forwarding to other nodes known to be receivers of the source is allowed in *multicast sender mode* (illustrated in Figure 14). ASTM allows sources to multicast data directly to a receiver member without being forced to travel to the RP, if the sources are nearby. This method is called *adaptive multicast* (adaptive persource multicast routing), and is depicted in Figure 14. Receivers can elect to receive packets sent by a sender either from the RP-rooted shared tree or from the persource tree based on path length comparison. Switching between the shared tree and the persource tree based is accomplished by sending a *Join Request* with a forwarding list to the source to establish the forwarding path from the source to the receiver and letting the record for the source-receiver pair expire in the forwarding list on the *Join Requests* to the RP. When nodes move and the path becomes

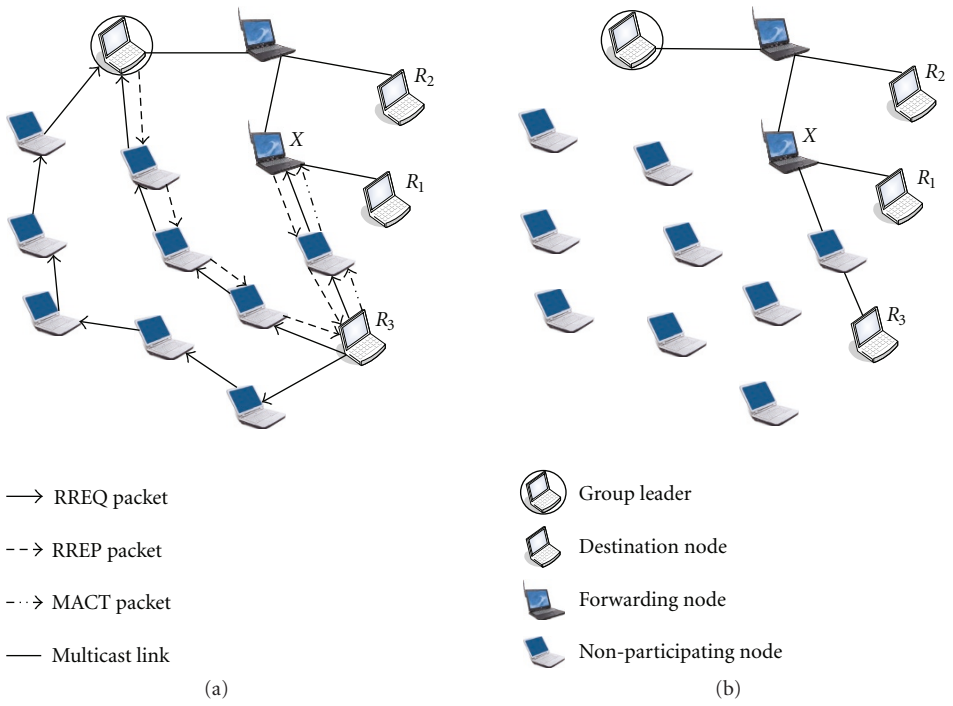


FIGURE 12: (a) Node R_3 joining the multicast tree. (b) Multicast tree at the end of the joining process.

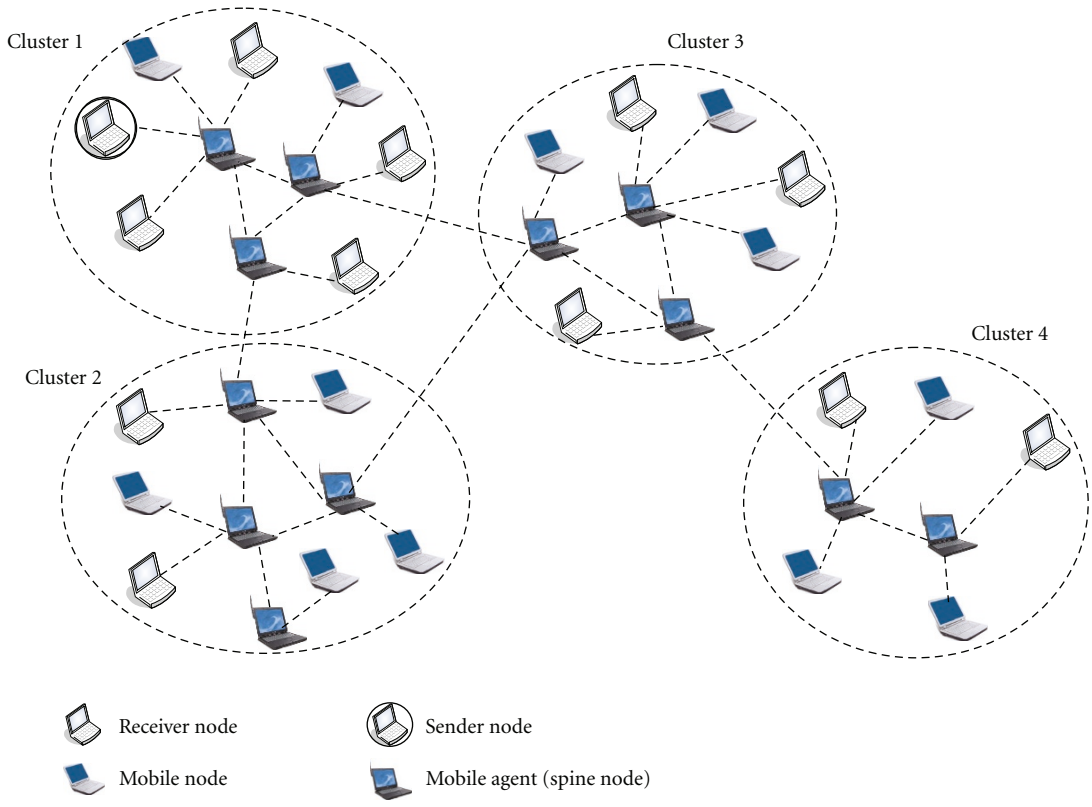


FIGURE 13: A spine based ad hoc network with 4-clusters. © IEICE 2001.

much longer than the distance from the receiver R_j to the RP, then the receiver R_j can switch back to the shared forwarding tree rooted at the RP.

Discussion. ASTM has a single point of failure, since it is based on the RP. Moreover, as mobility increases, throughput decreases, due to the inability of the routing and multicast protocol to keep up with node movements. In the case of adaptive multicast, there may be packets traveling from a source, say X, to a destination, say Y, on paths which are much longer than the shortest path between the source X and the destination Y. This may lead to an efficiency problem.

Ad Hoc Multicast Routing Protocol Utilizing Increasing ID Numbers (AMRISs)

Protocol Description. AMRIS [13] 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*. A multicast delivery tree rooted at a particular node with the smallest *msm-id*, called the Sid, is constructed. *msm-id* increases as the tree expands from the source (generally, the Sid is the source if there is only one sender for a group). In the case of multiple senders, the sender with smallest *msm-id* is selected as the Sid. A multicast session is initiated by a *NEW-SESSION* message sent by the Sid. The *NEW-SESSION* message includes the Sid's *msm-id* and the routing metrics. Neighbor nodes receiving this message generate their own *msm-id*, which is larger than that specified in the *NEW-SESSION* message. The nodes rebroadcast the *NEW-SESSION* message with their own *msm-ids*. To join a multicast group, a node sends a *Join Request* (*JREQ*) to the parent node with smallest *msm-id*. If the parent is a member in the desired multicast group, it sends a *Join Acknowledgment* (*JACK*). Otherwise, the parent sends a *JREQ* to its parent. Figure 15 illustrates the joining process in AMRIS. When a link between two nodes breaks, the node with the larger *msm-id* is responsible for rejoining. A node attempts to rejoin the tree by executing *Branch Reconstruction* (BR), which has two main subroutines, BR_1 and BR_2. However, BR_1 is executed when the node has neighboring potential parent nodes which it can attempt to join, and BR_2 is executed when the node does not have any neighboring nodes that can be potential parents.

Discussion. AMRIS repairs the broken links by performing local route repair without the need for any central controlling node, thereby reducing the control overhead. Introducing the concept of ID number avoids loop formation. However, AMRIS acts after a link has already failed, and so it introduces a significant delay in route recovery and packet loss. In addition, nodes periodically send beacons to signal their existence. As a result, bandwidth is wasted and also many packets are lost due to collisions between beacons.

On-Demand Multicast Routing Protocol (ODMRP)

Protocol Description. ODMRP [35] is a source-initiated multicast routing protocol. It introduces the concept of *forwarding group* (only a subset of nodes forwarding the multicast packets). When multicast sources have data to send but do not have routing or membership information, they flood a *JOIN DATA* packet. When a node receives a nonduplicate *JOIN DATA* packet, it stores the upstream node ID and rebroadcasts the packet. When the *JOIN DATA* packet reaches a multicast receiver, the receiver creates a *JOIN TABLE* packet and broadcasts to the neighbors. When a node receives a *JOIN TABLE* packet, it checks whether or not the next node ID of one of the entries matches its own ID. If it does, the node realizes that it is on the path to the source and thus is part of the forwarding group. It then broadcasts its own *JOIN TABLE* packet built upon matched entries. The *JOIN TABLE* packet is thus propagated by each forwarding group member until it reaches the multicast source via the shortest path. Figure 16 illustrates the joining process. This process constructs (or updates) the routes from sources to receivers and builds a mesh of nodes, the *forwarding group*. Multicast senders refresh the membership information and update the routes by sending *JOIN DATA* packets periodically. No explicit control message is required to leave the group. Any node which needs to leave the group just stops sending the *JOIN DATA* packet, or, if it does not need to receive from the multicast group, it does not send the *JREP* packet. Simulation results have shown that mesh-based protocols significantly outperform tree-based protocols. In addition, compared with another mesh protocol CAMP, ODMRP produced less control overhead and efficiently utilized those control packets to deliver more data packets to multicast members.

PatchODMRP [37] is proposed to save control overhead introduced by ODMRP by utilizing local route maintenance (3-hop). In spite of this modification, its local route maintenance is still considerable. In order to further reduce the scope of that maintenance and incur less control overhead, PoolODMRP is proposed [21]. With the aid of pool node technology and by reducing the scope of local route maintenance to 1-hop, PoolODMRP reduces its control overhead greatly.

In order to alleviate the limitations of PoolODMRP (it is less efficient in local route maintenance than PatchODMRP, as it consumes CPU resources to collect route information from data packets and uses the BEACON signal from the MAC layer to maintain the status of forwarding nodes), an ad hoc multicast protocol based on passive data acknowledgement, called PDAODMRP, has been proposed [31]. PDAODMRP knows the status of its downstream forwarding nodes by route information collected from data packets instead of by means of the BEACON signal of the MAC layer, and reduces the wasting of wireless bandwidth created by the BEACON signal. It has also adopted a new method of route information collection from data packets to reduce CPU usage. In addition, it has adopted dynamic local route maintenance to enforce its local route maintenance methodology.

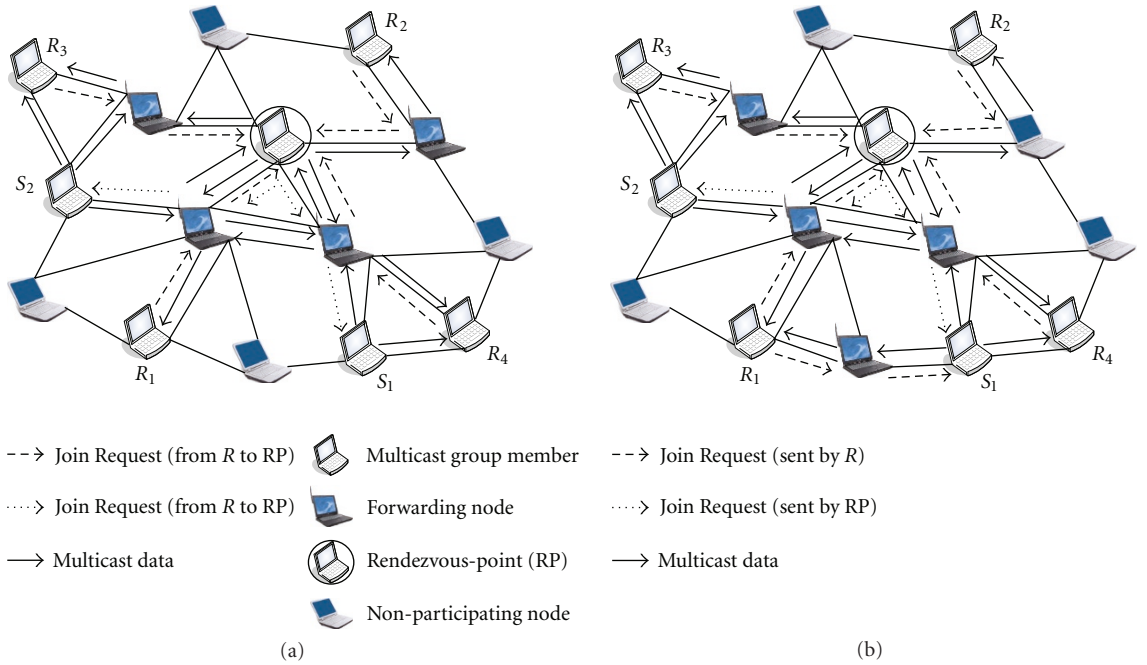


FIGURE 14: Multicast protocols in ASTM: (a) multicast senders, (b) adaptive multicast. © IEEE 1998.

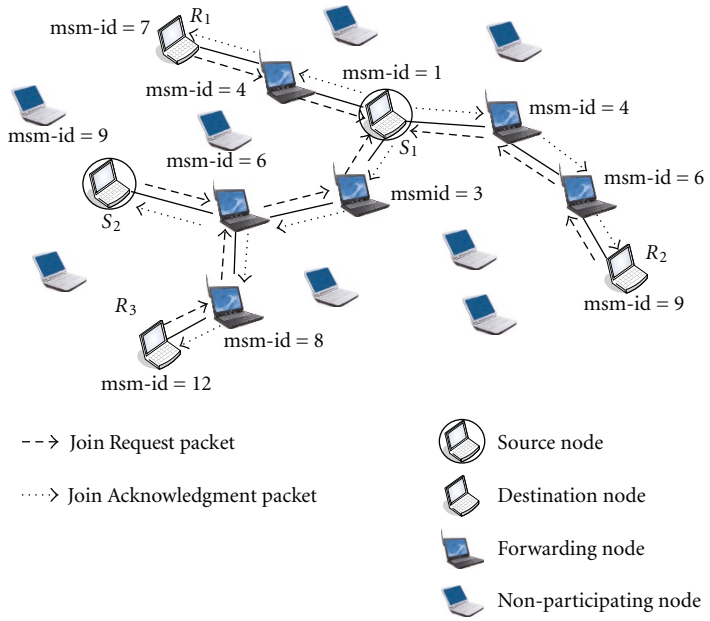


FIGURE 15: Joining process in AMRIS.

Another variation of ODMRP is E-ODMRP [60] (Enhanced ODMRP). E-ODMRP enhances ODMRP with an adaptive route refresh mechanism based on receiver reports on link breakages, rather than on mobility prediction. In particular, the enhancement changes the route refreshing period dynamically to reduce the flooding overhead of *JOIN QUERY* packets. In this way, it improves the efficiency of the protocol. In addition, E-ODMRP proposes a local route recovery mechanism based on ERS. However, this scheme incurs additional control packets (i.e., the *RECEIVER JOIN* packet) and requires additional processing at nodes, which may not be available in low end mobile devices. Furthermore, malicious or misbehaving nodes can drain the resources of multicast receivers and forwarding nodes by initiating frequent ERSs. Simulation results show that E-ODMRP reduces packet overhead by up to 50%, while keeping the

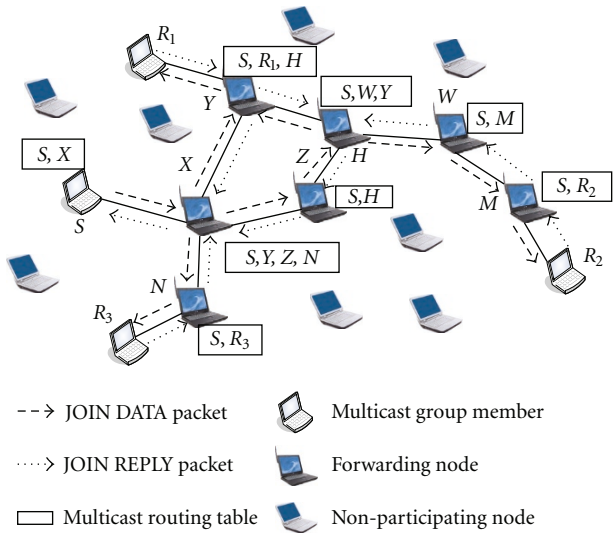


Figure 16: Joining process in ODMRP.

packet delivery ratio similar to that of the original ODMRP. Moreover, the simulation results also confirm that E-DMRP outperforms ADMR [15].

Discussion. The main disadvantage of ODMRP is high control overhead while maintaining current forwarder groups and all network request package flooding. This problem can be overcome using preemptive route maintenance, as suggested by Xiong et al. [61]. Another disadvantage is that the same data packet propagates through multiple paths to a destination (duplicate packets), which reduces multicast efficiency. In addition, ODMRP has a scalability problem. Finally, the sources must be part of the group's multicast mesh, even when they are not interested in receiving multicast packets.

Adaptive Core Multicast Routing Protocol (ACMRP)

Protocol Description. ACMRP [14] is an on-demand core-based multicast routing protocol. A multicast mesh is shared by the sources of a group. A designated node, called a *core*, while not well known, adapts to the current network topology and group membership status. A multicast mesh is created and maintained by the periodic flooding of a *Join Request* packet which is performed by the adaptive *core*. When a node receives a fresh *JREQ*, it inserts the packet into its *jreq cache* and updates the route to the *core*. Then, it changes the "upstream node address" field in the packet to its own address and retransmits the packet. Group members (including multicast receivers as well as sources) send a *Join Reply* (*JREP*) packet to their upstream node on receipt of a nonduplicate *JREQ* packet. Upon receiving the *JREP*, the upstream node stores the group address, which will be used to forward multicast packets destined for the group in the future. This node is called a forwarding node. It inserts a (*group address, source address*) pair into the *forwarding group table*. Then, it sends a *JREP* to its own upstream node. Eventually, the *JREP* reaches the *core*. The backward propagations of *JREPs* construct multicast routes between group members and the core. Consequently, a multicast mesh is established. The adaptive *core* mechanism of ACMRP automatically handles any link failure, node failure, or network partition. Figure 17 shows an example of multicast mesh creation and packet delivery. *Core* broadcasts a *JREQ*, and group members (S_1, S_2, R_1, and R_2) send *JREP*s to their upstream nodes (resp., *X*, *Core*, *Y*, and *Core*). As a result, intermediate nodes (*X* and *Y*) and *Core* become forwarding nodes. As shown in Figure 17(b), a multicast mesh provides alternative multicast routes. Even if the link between *A* and *Core* is broken, the packet is transferred to R_2 via $S_1 \rightarrow X \rightarrow Y \rightarrow Core \rightarrow R_2$. Simulations have shown that ACMRP performs well with less control traffic overhead compared to ODMRP [35].

Discussion. The enhanced adaptivity of ACMRP minimizes core dependency, thereby improving performance and robustness and making ACMRP operate well in dynamically changing networks. An ACMRP scales well to large numbers of group members and is suitable even in a heavily loaded ad hoc network. One disadvantage of this protocol is that the paths between the sources and the receivers are not optimal. Also, the selection of the core is critical. The position of the core node is very important. It should be placed with the minimum hop counts of routes toward group members and guarantee that it has enough residual power for support until the next core is elected.

Dynamic Core-Based Multicast Routing Protocol (DCMP)

Protocol Description. DCMP [24] is an extension to ODMRP [35] and attempts to reduce the number of senders flooding *JREQ* packets by selecting certain senders as cores. This reduces the control overhead and therefore improves the efficiency of the ODMRP multicast protocol. DCMP constructs a mesh similar to that in ODMRP. It reduces the number of sources flooding the *JREQ* by having three types of sources: active, passive, and core active. Only active sources and core active sources flood the *JREQ*. Packets initiated at passive sources are sent to the core active node (as a proxy for passive sources), which forwards them to the mesh. The number of passive sources a single core active source can serve must be limited for robust operation. The distance (number of hops) between a passive source and its core active node must also be limited to ensure that the packet delivery ratio is not reduced. Figure 18 reveals the mesh construction in DCMP, where the parameters *MaxHop* and *MaxPassSize* (the maximum number of passive sources, a limitation that allows the mesh to have enough forwarding nodes for robust operation) are set to two. Since S_2 and S_3 are at a hop distance of 2 from each other (which is equal to *MaxHop*), S_3 becomes passive and uses a proxy in the core active node S_2. No other pair of sources is separated by a hop distance of less than 2, and so eventually S_1 becomes the active source, S_3 a passive source, and S_2 a core active source. The number of forwarding nodes is reduced, as compared to ODMRP [35], without much reduction in robustness or packet delivery ratio.

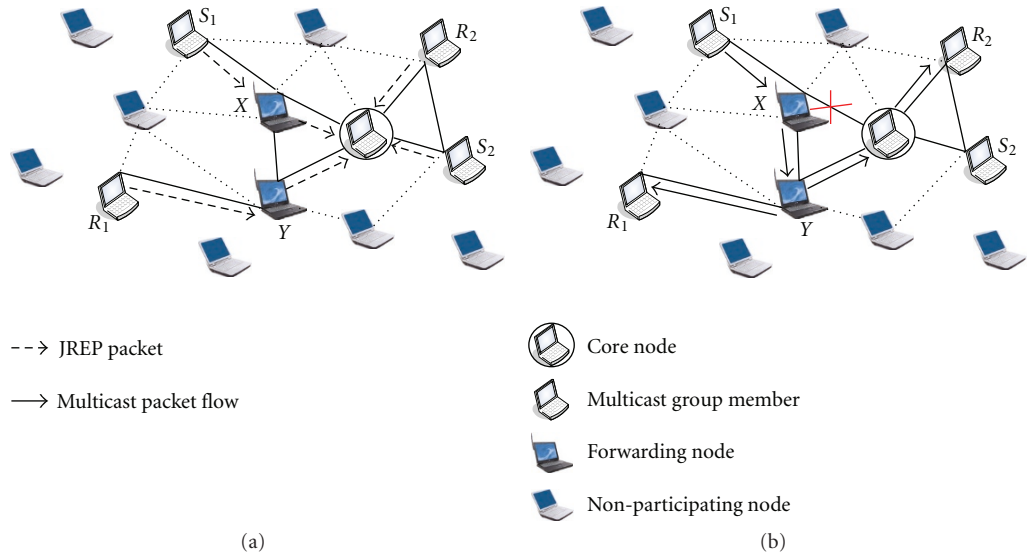


FIGURE 17: Multicast mesh in ACMRP: (a) the propagation of *JREP*, (b) multicast packet deliveries from S_1 to R_1 and R_2 on a link failure. © Kluwer Academic Publishers 2003.

Discussion. DCMP does not entirely alleviate the drawback of ODMRP, which is multiple control packet floods per group, but it is still much more scalable than ODMRP. It also has a high delivery ratio compared to ODMRP. However, in the case of failure of the core active source, multiple multicast sessions will fail.

Multicast for Ad Hoc Networks with Swarm Intelligence (MANSI)

Protocol Description. MANSI [12] applies swarm intelligence mechanisms to the problem of multicast routing in MANETs. Swarm intelligence refers to complex behaviors that arise from very simple individual behaviors and interactions, which are often observed in nature, especially among social insects such as ants and honey bees. Although each individual (an ant, e.g.,) has little intelligence and simply follows basic rules using local information obtained from the environment, global optimization objectives emerge when ants work collectively as a group. Similarly, MANSI utilizes small control packets which deposit information at the nodes they visit. This information is used later by other control packets. MANSI adopts a core-based approach to establish multicast connectivity among members through a designated node (core). The core is the first node that initiates the multicast session. It announces its existence to the others by flooding the network with a *CORE ANNOUNCE* packet. Each member node then relies on this announcement to reactively establish initial connectivity by sending a *JREQ* back to the core via the reverse path. Nodes receiving a *JREQ* addressed to themselves become forwarding nodes of the group and are responsible for accepting and rebroadcasting nonduplicated data packets, regardless of from which node the packets were received. To maintain connectivity and allow new members to join, the core floods *CORE ANNOUNCE* periodically, as long as there are more data to be sent. As a result, these forwarding nodes form a mesh structure that connects the group members, while the core serves as a focal point for forwarding set creation and maintenance. Figure 19 illustrates the initialization of the multicast tree. MANSI tries to reduce the number of nodes used to establish connectivity. For this purpose, nodes tend to choose paths that are partially shared by others to reduce the size of the forwarding set. Periodic exploration messages are deployed by members to search for new forwarding nodes with lower cost. Active forwarding members reply to these search packets. If the cost of the new path is lower for the intermediate and requesting nodes, the requester switches to the new route and the old one expires.

Discussion. MANSI adopts the concept of swarm intelligence to reduce the number of nodes used to establish multicast connectivity. However, the path between the multicast member and forwarding set to the designated core is not always the shortest. MANSI employs a mesh-based approach to increase redundancy by allowing packets to be forwarded over more than one path, thereby raising the chances of successful delivery. In MANSI, group connectivity can be made more efficient by having some members share common paths to the core with other members in order to further reduce the total cost of forwarding data packets. Since a node's cost is abstract and may be defined to represent different metrics, MANSI can be applied to many variations of multicast routing problems for ad hoc networks, such as load balancing, secure routing, and energy conservation.

Forward Group Multicast Protocol (FGMP)

Protocol Description. FGMP [26] is a multicast routing protocol that creates a multicast mesh on demand, and

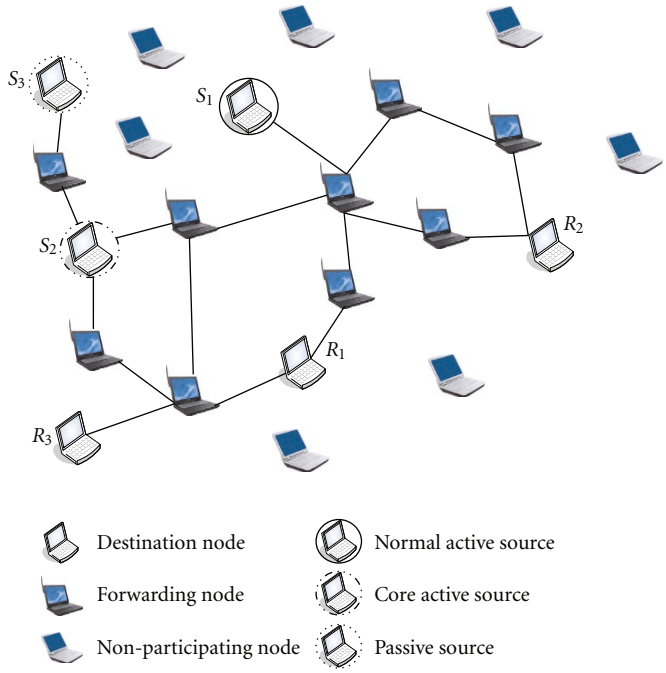


Figure 18: Multicast mesh in DCMP. © ACM 2002.

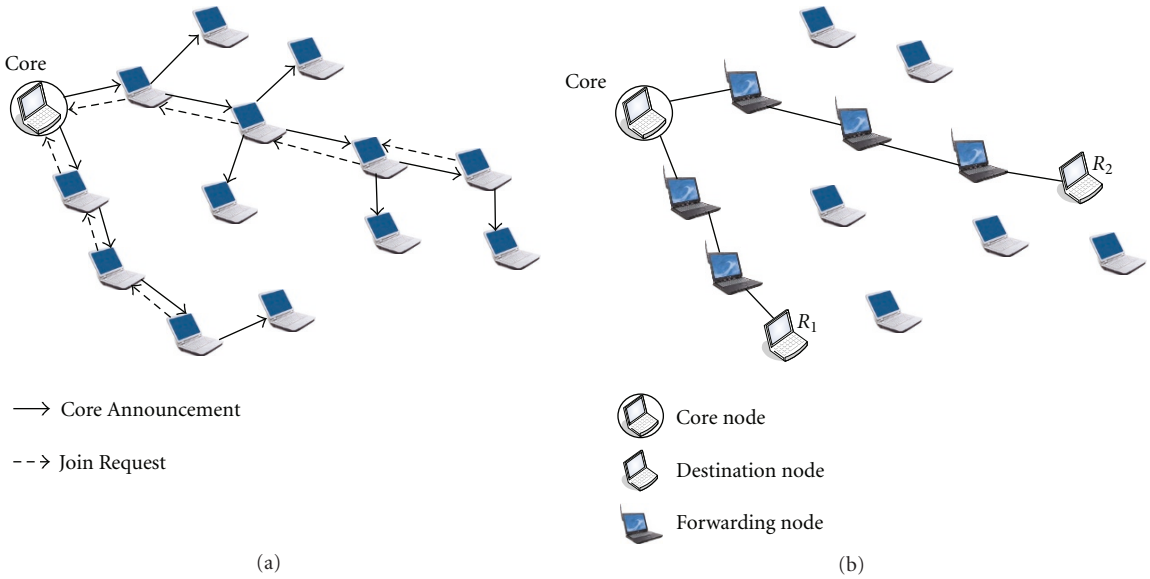


Figure 19: Multicast initialization in MANSI: (a) the core node sends CORE ANNOUNCMNET, (b) R_1 and R_2 joining the multicast group.

is based on the forwarding group concept. FGMP keeps track not of links but of groups of nodes which participate in multicast packet forwarding. A forwarding group FG is associated with each multicast group G. Any node in FG is in charge of forwarding (broadcast) multicast packets of G. That is, when a forwarding node (a node in FG) receives a multicast packet, it will broadcast this packet if it is not a duplicate. All neighbors can hear it, but only neighbors that are in FG will first determine whether or not it is a duplicate and then broadcast it in turn. There are two ways to advertise the membership, a sender advertising (FGMP-SA) approach or a receiver advertising (FGMP-RA) approach. In FGAP-RA, each receiver periodically floods its membership information by *JREQ*. When a sender receives the *JREQ* from receiver members, it updates its member table with all receivers in the group. In FGAP-SA, senders periodically flood the sender information to announce their presence in the network. Receivers will collect senders' status, and then periodically broadcast *joining tables* to create and maintain the forwarding group FG. The *joining table* has the same

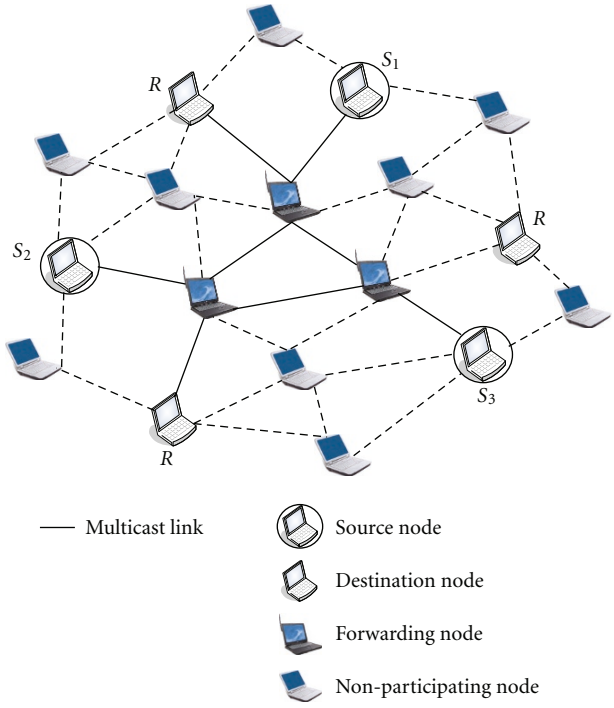


FIGURE 20: Multicast tree of FGMP. © Kluwer Academic Publishers 1998.

format as the *forwarding table* except that the joining table contains the sender *IDs* while the forwarding table contains receiver *IDs*. Forwarding flag and timer are set when a node receives the *joining table*. FG group is maintained (*Soft-State* refresh) by the senders in receiver advertising scheme and by the receivers in sender advertising scheme. Figure 20 shows an example of a multicast group containing three sources and three destinations. Forwarding nodes take the responsibility of forwarding multicast packets.

Discussion. FGMP limits flooding within the selected FG nodes, thereby reducing channel and storage overhead. In a high mobility environment, frequent FG changes can adversely affect the protocol's performance. FGMP provides a feasible solution only in small networks and when the number of senders is greater than the number of receivers. It is more efficient to utilize FGMP-SA when the number of sources is smaller than the number of destinations in the multicast group. However, when the number of sources is greater than the number of destinations, FGMP-RA is more efficient than FGMP-SA.

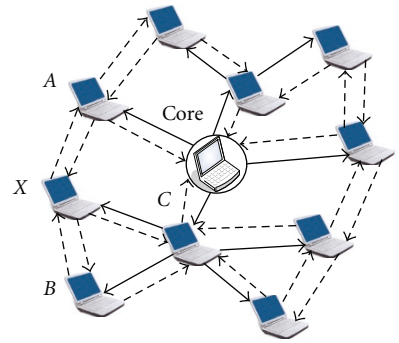
Protocol for Unified Multicasting through Announcements (PUMAs)

Protocol Description. PUMA [40] establishes and maintains a shared mesh for each multicast group. It eliminates the need for a unicast routing protocol or the preassignment of cores (it makes use of dynamic cores) to multicast groups. PUMA uses a receiver-initiated approach, in which receivers join a multicast group using the address of a core node, without the need for network-wide flooding of control or data packets from all the sources of a group. PUMA elects the first receiver of the group as the core of the group, and informs each node in the network of at least one next-hop to the elected core of each group. A core node of a group transmits multicast announcements periodically for that group. As the multicast announcement travels through the network, it establishes a *connectivity list* at every node in the network. Figure 21 illustrates the propagation of multicast announcements and the building of *connectivity* lists. Using these lists, nodes are able to establish a mesh and route data packets from senders to receivers. Every receiver connects to the elected core along all the shortest paths between the receiver and the core. When a receiver wishes to join a multicast group, it first determines whether or not it has received a multicast announcement for that group before. If the node knows the core, it starts transmitting multicast announcements and specifies the same core for the group. Otherwise, it considers itself the core of the group and starts transmitting multicast announcements periodically to its neighbors, stating itself as the core of the group. When a node wishes to send data to a group, it forwards the data packets to the node from which it has received the best multicast announcement (the one with the higher ID). A node forwards a multicast data packet it receives from its neighbor if the neighbor's parent is the node itself. Hence, multicast data packets move hop by hop, until they reach mesh members. The packets are then flooded within the mesh, and group members use a packet ID cache to detect and discard packet duplicates. Like other multicast muting protocols using sequence numbers, PUMA needs to recycle sequence numbers and handle failures that cause a core to reset the sequence number assigned to a multicast group. The sequence number of a multicast announcement is only increased by the core of the group.

Discussion. PUMA minimizes data packet overhead by using only one node, that is, the core node floods the network. In addition, it tends to concentrate mesh redundancy in the region where receivers exist by including all the shortest paths from the receivers to the core, which is also a receiver.

CAMP: Core-Assisted Mesh Protocol

Protocol Description. CAMP [22] extends the notion of core-based trees CBT [62] introduced for Internet multicasting into multicast meshes, which have much richer connectivity than trees. A shared multicast mesh is defined for each multicast group to maintain the connectivity of multicast groups, even during the frequent movement of network routers. CAMP establishes and maintains a multicast mesh, which is a subset of the network topology, which provides multiple paths between a source-receiver pair and ensures that the shortest paths from receivers to sources (called reverse shortest paths) are part of a group's mesh. One or multiple cores are defined per multicast group to assist in join operations; therefore, CAMP eliminates the need for flooding. CAMP uses a receiver-initiated approach for



Connectivity list at node X		
Neighbor	Multicast announcement	
	Distance to core	Parent
A	1	Core
B	2	C
C	1	Core

⟶ Indicate the neighbor from which a node receives its best multicast announcement

Core node

Figure 21: Propagation of multicast announcements. © IEEE 2004.

receivers to join a multicast group. A node sends a *JREQ* toward a core if none of its neighbors is a member of the group; otherwise, it simply announces its membership using either reliable or persistent updates. If cores are not reachable from a node that needs to join a group, the node broadcasts its *JREQ* using an ERS, which eventually reaches some group member. In addition, CAMP supports an alternate way for nodes to join a multicast group by employing simplex mode. Figure 22 shows a multicast mesh in CAMP.

Discussion. CAMP needs an underlying proactive unicast routing protocol (the Bellman-Ford routing scheme) to maintain routing information about the cores, in which case considerable overhead may be incurred in a large network. Link failures have a small effect in CAMP, so, when a link fails, breaking the reverse shortest path to a source, the node affected by the break may not have to do anything, because the new reverse shortest path may very well be part of the mesh already. Moreover, multicast data packets keep flowing along the mesh through the remaining paths to all destinations. However, if any branch of a multicast tree fails, the tree must reconnect all components of the tree for packet forwarding to continue to all destinations.

Source Routing-Based Multicast Protocol (SRMP)

Protocol Description. SRMP [42] is an on-demand multicast routing protocol. It constructs a mesh topology to connect each multicast group member, thereby providing a richer connectivity among members of a multicast group or groups. To establish a mesh for each multicast group, SRMP uses the concept of FG nodes. SRMP applies the source routing mechanism defined in the Dynamic Source Routing (DSR) [63] protocol to avoid channel overhead and to improve scalability. Also, SRMP addresses the concept of connectivity quality. Moreover, it addresses two important issues in solving the multicast routing problem: the path availability concept and higher battery life paths. When a source node that is not a group member wishes to join the group, it broadcasts a *JREQ* packet to its neighbors, invoking a route discovery procedure toward the multicast group. The *JREQ* packet contains the ID of the source node, the multicast group ID, and a *Sequence number* field. The *Sequence number* is set by the source node for each *JREQ* packet generated, and is used to detect packet duplication. A first multicast receiver receives the *JREQ* packet, stores the multicast routing information, and then checks its *Neighbor Stability Table* for stability information among its neighbors (association stability, link signal strength, and link availability). Battery life is also verified considering the power needed to transmit to each neighbor. A neighbor is selected as an FG node if the four selection metrics satisfy their predefined thresholds. Then, the receiver starts sending a *JREP* packet to each selected node, setting its type as "member node" in the *Neighbor Stability Table*. If there are no neighbor nodes satisfying the predefined thresholds, the node with the best metrics among all the neighbors will be selected as an FG node. Once the route is constructed, a multicast source node starts sending the multicast data toward the multicast group members. Any node wishing to leave the multicast group sends *Leave-Group* messages to its neighbor members. Figure 23 shows the multicast mesh in SRMP.

Discussion. SRMP selects the most stable paths among multicast group members. This maximizes the lifetime of the routes, offers more reliability and robustness, and results in the consumption of less power. In addition, it discovers routes and detects link failures on demand, thereby minimizing channel and storage overhead (improving the scalability of the protocol), as well as saving bandwidth and network resources. The value of the four metrics used in selecting the paths may not be globally constant, however. They probably vary with different network load conditions. So, the four metrics must be made to be adaptive to the network load conditions.

Neighbor-Supporting Multicast Protocol (NSMP)

Protocol Description. NSMP [33] is a source-initiated multicast routing protocol, and is an extension to ODMRP

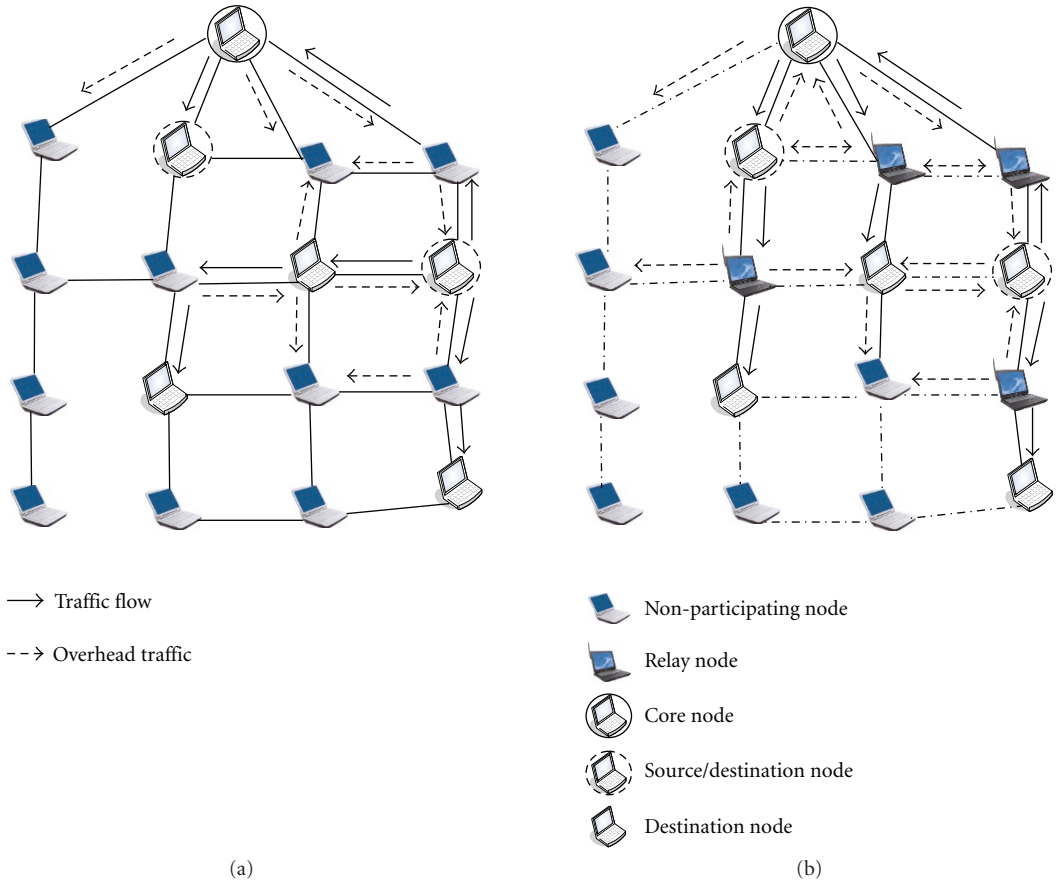


Figure 22: Multicast mesh in CAMP: (a) traffic flow from node X, (b) equivalent multicast shared tree. © IEEE 1999.

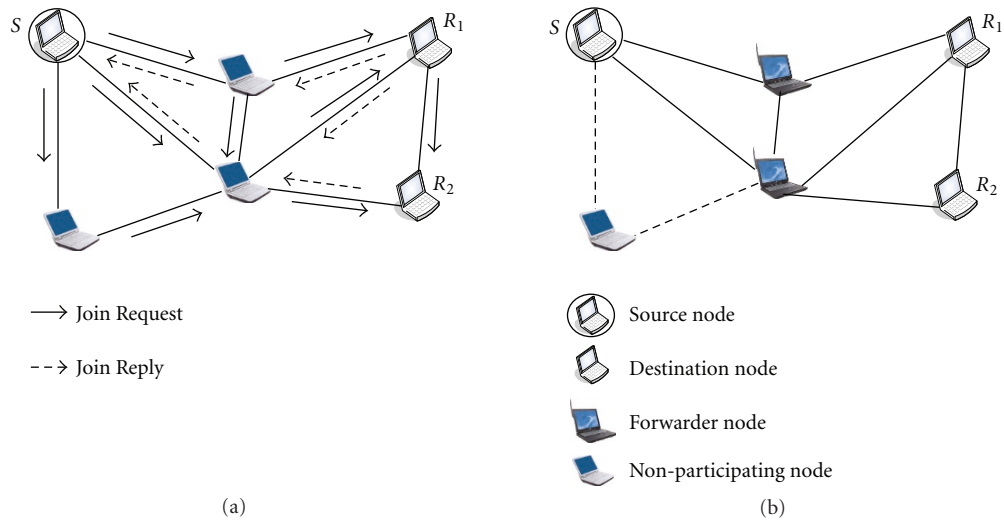


Figure 23: Multicast mesh in SRMP: (a) mesh initialization, (b) mesh creation.

[35]. A mesh is created by a source, which floods a request throughout the network. Intermediate nodes cache the upstream node information contained in the request and forward the packet after updating this field. When any receiver node receives the route discovery packets, it sends replies to its upstream nodes. Intermediate nodes receiving these replies make an entry in their routing tables and forward the replies upstream toward the source. In the case where the receiver receives multiple route discovery packets, it uses a relative weight metric (which depends on

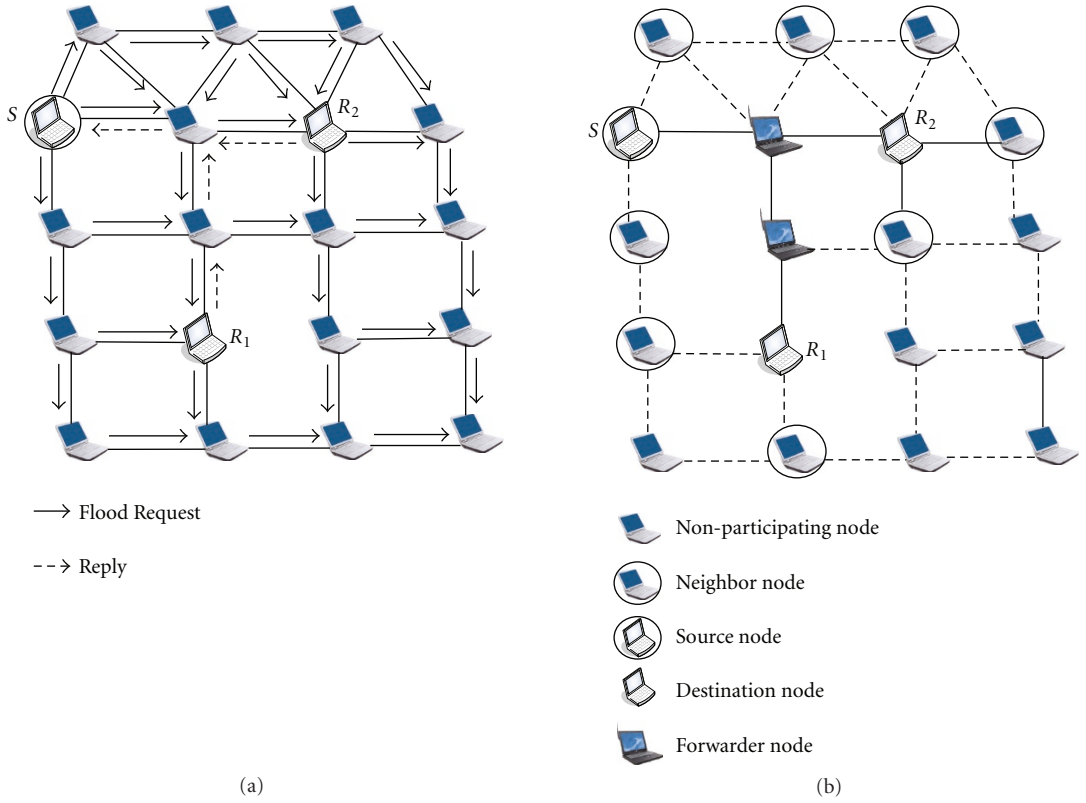


Figure 24: Multicast mesh in NSMP: (a) mesh initialization, (b) mesh creation. © IEEE 2000.

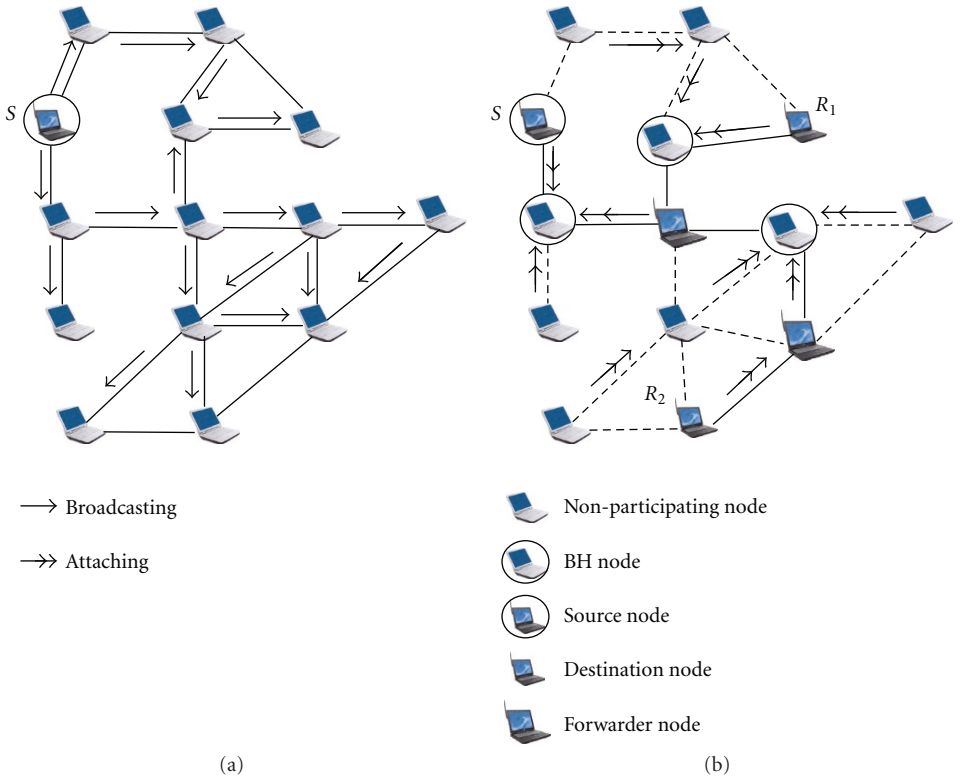


Figure 25: BH selection in a multicast region: (a) broadcasting, (b) selecting BHs, attaching NBHs to BHs and determining multicast routes. © Elsevier 2006.

the number of forwarding and nonforwarding nodes on the path from the source to the receiver) for selecting one of the multiple routes. A path with the lowest value of relative weight is chosen. Figure 24 illustrates how a multicast mesh is built. In order to maintain the connectivity of the mesh, the source employs local route discoveries by periodically transmitting local requests, which are only relayed to mesh nodes and multicast neighbor nodes (nodes that are directly connected to at least one mesh node) to limit flooding, while keeping the most useful nodes informed. Any new receiver wanting to join the multicast group must wait for one of these local requests to join the desired multicast group. Replies are sent back to the source to repair broken links. Only nodes away from the source by two hops or less can join the mesh with a local request. Otherwise, they have to flood the member request.

Discussion. NSMP is aimed at reducing the flood of control packets to a subset of the entire network. It utilizes node locality to reduce control overhead while maintaining a high delivery ratio. NSMP favors paths with a larger number of existing forwarding nodes to reduce the total number of multicast packets transmitted. It is preferable to make the relative weight metric adaptive to variations in the network load conditions.

On-Demand Global Hosts for Ad Hoc Multicast (OGHAM)

Protocol Description. OGHAM [34] constructs a two-tier architecture by selecting backbone hosts (BHs) on demand for multicast services. Each multicast member must be attached to a BH. Hosts with a minimal number of hops to the other hosts, rather than those with a maximal number of neighbors, will be adopted as BHs in order to obtain shorter multicast routes. BHs are responsible for determining multicast routes, forwarding data packets, handling dynamic group membership (the nodes can dynamically join or leave the group), and updating multicast routes due to host movement. When a source S wishes to create a multicast group, it first tries to find a BH within a region with a radius of $2r$-*hops* ($r \geq 1$ is a predefined integer) centered at S. If such a BH can be found, then S is attached to it. Otherwise, S broadcasts a message in a larger region, called a multicast region, with a radius of γ-hops ($\gamma \geq 2r$ is a predefined integer) centered at S for collecting neighboring information. Upon receiving the message, hosts in the multicast region reply their neighboring information to S. With this information, S then selects BHs and attaches neighbor BHs (NBHs) to BHs. After BHs in the multicast region are selected, receiver nodes can join the multicast group by asking the attached BHs to query the location of the source. The BH attached to the source then replies to the queries. Through round-trip communication (querying and replying), the BH attached to the source can determine the multicast routes from the source to the receiver nodes. This is depicted in Figure 25. If a node outside a multicast region 1 in Figure 26 is attempting to join the multicast group created by the source node S in Figure 25(b), it creates a multicast region (see Figure 26). There are two BHs (BH_4 and BH_5) selected in the new multicast region and R_3 is attached to BH_4. In order to locate the source node S, BH_4 floods a message. Upon receiving the message, BH_2 replies to R_3. Through the message exchange, a multicast route ($S \rightarrow BH_2 \rightarrow FN_1 \rightarrow BH_1 \rightarrow FN_2 \rightarrow FN_4 \rightarrow BH_4 \rightarrow R_3$) from S to R_3 is then determined.

Discussion. OGHAM minimizes transmission time and lost packets because BHs minimize the total number of hops to all hosts (receivers). In OGHAM, once the infrastructure for a particular multicast group has been constructed, the selected BHs are globally available for the other ad hoc multicast groups. Therefore, it is not necessary for follow-up multicast groups to flood again in order to construct additional infrastructures. Hence, as the group size or the group number increases, the ratio of control packets declines (very scalable).

Fireworks: An Adaptive Multicast/Broadcast Protocol

Protocol Description. Fireworks [64] is a hybrid 2-tier multicast/broadcast protocol that adapts to maintain performance, given the dynamics of the network topology and group density. It creates a cohort of broadcast (lower tier) distribution in areas with many members, while it develops a multicast backbone (upper tier) to interconnect these dense pockets (see Figure 27). The multicast tree is constructed as follows. When a node wishes to join a multicast group, it broadcasts an *ADVERTISE* message to its 2-hop neighborhood. Upon reception of a unique *ADVERTISE* message, nodes update their joining group table, as per the message contents. Following this (discovery) phase, each joining node would have obtained the 2-hop local topology information. This information may be used during the decision phase, in which, if the joining node receives multiple *LEADER* messages, it will pick the cohort leader with the shortest distance and highest cohesiveness (a state variable that maintains the affinity of group members within a node's 2-hop neighborhood). It then joins the cohort leader by unicasting a *CHILD* message containing its *address*, *mcast-address*, and *hop-count* to the selected cohort leader. If there are two or more cohort leaders with the same distance and cohesiveness, the joining node will select the cohort leader with the highest *nodeID*. If the joining node does not receive any *LEADER* message, then it elects itself as a cohort leader and serves a cohort. After that, it broadcasts a *LEADER* message to its 2-hop neighborhood so as to notify them of the presence of a new cohort leader. At the end of these phases, the lower tier is created. To enable the creation of the upper tier, the multicast source periodically broadcasts a *SOURCE-QUERY* message to the network. Intermediate nodes forward unique *SOURCE-QUERY* messages further. When a cohort leader receives the *SOURCE-QUERY* message, it unicasts a *SOURCE-REPLY* message back to the source via the reverse path. The nodes along the unicast path toward the source become the forwarding nodes for the multicast group. A cohort member could leave a multicast group anytime by

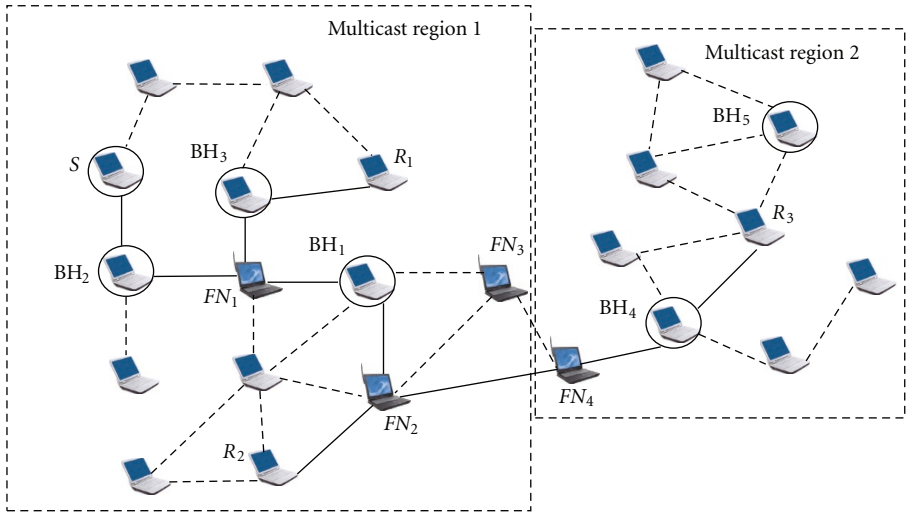


Figure 26: Determining multicast routes across two multicast regions. © Elsevier 2006.

stopping unicasting of the *CHILD* message to its cohort leader. If a cohort leader decides to leave the multicast group, it stops transmitting the *LEADER* message. Cohort members, upon discovering the absence of a leader, will perform the joining and discovery phases as described before.

Discussion. Fireworks significantly reduces the protocol overhead by exploiting the broadcast nature of the mobile ad hoc network in the area with many group members (cohort). Moreover, since Fireworks employs broadcasting within a cohort, the inherent redundancy provides reliability and packet delivery performance that is comparable with that of ODMRP [35]. Fireworks develops a multicast backbone (Tree-based) to interconnect the dense pocket, which means that a link failure could affect multiple paths and, therefore, reduce the packet delivery ratio and introduce more overhead as well, especially in a highly dynamic environment. Another disadvantage is that Fireworks depends on the 2-hop local topology information during the decision phase, therefore, in the case of packet loss, a reduction in the accuracy of the topology information could affect the performance of Fireworks.

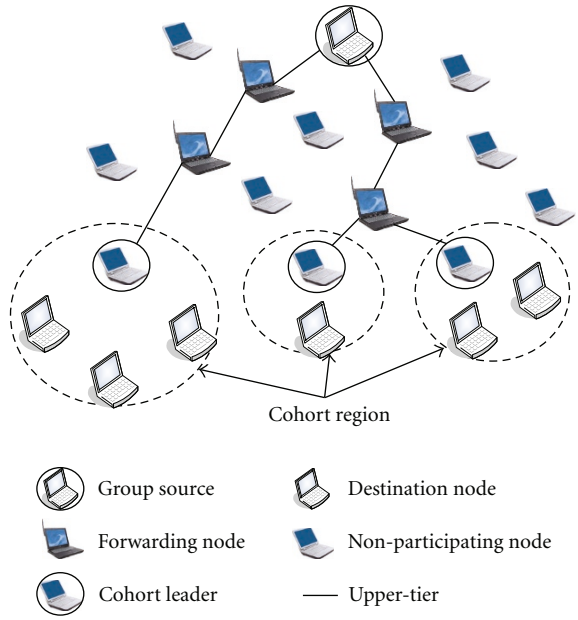


Figure 27: Fireworks 2-tier multicast hierarchy structure.

Agent-Based Multicast Routing Scheme (ABMRS)

Protocol Description. ABMRS [65] employs a set of static and mobile agents in order to find the multicast routes, and to create the backbone for reliable multicasting, as a result of which the packet delivery ratio is improved. The steps of the ABMRS are the following: reliable node identification, reliable node interconnection, reliable backbone construction, multicast group creation, and network and multicast group management. The Route Manager Agent (RMA) at each node computes the Reliability Factor (RF, which depends on various parameters such as power ratio, bandwidth ratio, memory ratio, and mobility ratio) and advertises to each of its neighbors. The Network Initiation Agent (NIA) at each node receives the advertised packet and determines who has the highest RF. The node with the highest RF will announce itself as a reliable node and inform its RMA. The interconnection between the reliable nodes is illustrated in Figure 28. Consider that the reliable node X would like to find the path to the reliable node Z. The RMA of the reliable node X triggers the NIA. The NIA creates clones (agent cloning is a technique for creating an agent similar to that of the parent, where the cloned agent contains the code and information of the parent agent) and floods across the network. One of the NIA clones from node X will move to the reliable node Z through intermediate node Y and send the traced path back to the NIA of node X and destroy itself. Similarly, the other clones also send their traced path back to the NIA of node X and destroy themselves. Assume that

the NIA at node X decides that the path $X \rightarrow Y \rightarrow Z$ is the best (minimum hop path) to reach node Z. The NIA informs its RMA, the RMA of node Y, and the RMA of node Z. This information will be used to generate the forwarding table. After this step, the RMA in each of the reliable nodes will broadcast information about their adjacent reliable nodes throughout the network. Using this information, RMA applies Dijkstra's algorithm to compute the routes between the reliable nodes and generate the forwarding table. Intermediate nodes generate the forwarding table based on the information given by NIAs as described above. At the end of this step, the backbone is ready for communication. Finally, the multicast group is created by the Multicast Initiation Agent (MIA). MIA travels to each reliable node and invites the multicast group to join. After performing the initial membership survey and collecting the necessary group membership information, the MIA forms an initial multicast tree comprising reliable nodes, intermediate nodes, and group members. The Network Management Agent (NMA) is responsible for managing the multicast group. Whenever an intermediate node or reliable node is disconnected, the NMA will ask the RMA to initiate the NIA to find the new paths between the reliable nodes. A child node has the responsibility of finding a new reliable node whenever there is a disconnection between a reliable node and its child node because of mobility.

Discussion. ABMRS computes multicast routes in a distributed manner, which provides good scalability. ABMRS is more reliable, that is, it has a higher packet delivery ratio, than MAODV [29]. This is because ABMRS constructs the multicast tree based on reliable nodes. However, ABMRS incurs a significant control overhead compared to MAODV, especially when mobility and the multicast group size are increased. The reason for this is that more agents are generated to find a route to reliable nodes. ABMRS assumes the availability of an agent platform at all mobile nodes. However, in the case of agent platform unavailability, traditional message exchange mechanisms can be used for agent communication. As a result, more control overhead will be incurred. In addition, ABMRS is based on Dijkstra's algorithm for computing the routes between the reliable nodes, and, therefore, it needs to know the network topology in advance. As a result, it has a scalability issue, and a significant overhead will be incurred as well.

Optimized Polymorphic Hybrid Multicast Routing Protocol (OPHMR)

Protocol Description. OPHMR [66] is built using the reactive behavior of ODMRP [35] and the proactive behavior of the MZRP [32] protocol. In addition, the Multipoint Relay- (MPR-) based mechanism of the OLSR [67] protocol is used to perform an optimization forwarding mechanism. OPHMR attempts to combine the three desired routing characteristics, namely, hybridization (the ability of mobile nodes (MNs) to behave either proactively or reactively, depending on the conditions), adaptability (the ability of the protocol to adapt its behavior for the best performance when mobility and vicinity density levels are changed), and power efficiency. To enable hybridization and adaptability, that is, polymorphism, OPHMR introduces different threshold values, namely, power, mobility, and vicinity density. OPHMR is empowered with various operational modes which are either proactive or reactive, based on an MN's power residue, mobility level, and/or vicinity density level. In a route, each MN tries to determine the destination node according to its own strategy (proactive or reactive). Thus, the MNs try to find the next forwarding nodes by using their own routing tables, which are established in the background for proactive stations, or by using broadcasting for reactive stations. This feature ensures that any hysterical behavior is avoided. Each MN determines its mode of operation based on the threshold values mentioned earlier. When a node wants to join a multicast group or wants to send data to that group, it begins the route discovery procedure. If it is in reactive mode, it sends out a *JREQ* message and waits for replies. This is done as in ODMRP. If the node is in proactive mode or proactive ready mode, it first looks in its neighborhood table to see whether or not there are nodes that belong to the destination multicast group. If there are, it unicasts *JREQ* messages to all these nodes and waits for replies. Otherwise, the node will broadcast a *JREQ* message. When a node receives such a message and it is a member of the multicast group, it sends back a reply to the source of that message and updates its multicast routing tables to record the route. If the node could not send a reply, it checks its own behavior. If it is in reactive mode, it just propagates the *JREQ* message and records it in the route cache. If the node is in proactive mode or in proactive ready mode, it looks in its own neighborhood table to find the destination multicast group member. If there are members in its zone, it unicasts the *JREQ* message to all of them. If not, it just propagates the message. When the source node receives a reply, it updates its multicast routing tables to record the route and begins data transmission.

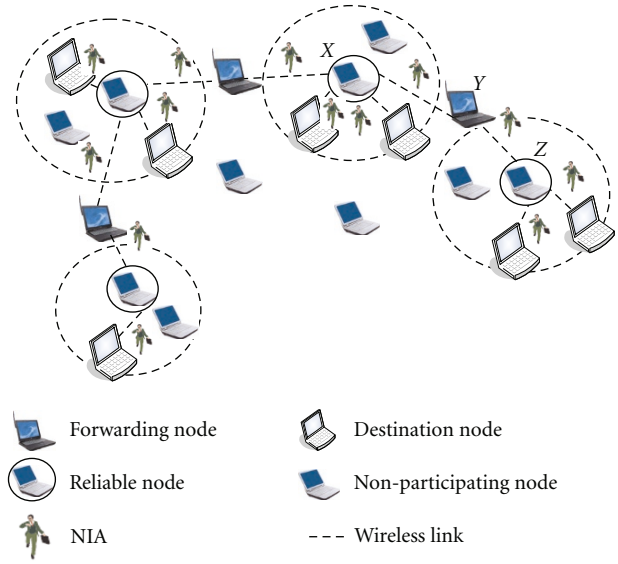


Figure 28: ABMRS multicast tree structure.

Table 2: Common pros and cons of IPLM.

Taxonomy Routing scheme + Multicast topology + Maintenance approach	Common pros/cons
Reactive + Source-Tree-based + Hard-State	(i) Loop-free (ii) High route acquisition latency (iii) Single point of failure (iv) Does not support QoS (v) Efficient traffic distribution (vi) Frequent link failure
Reactive + Shared-Tree-based + Hard-State	(i) Loop-free (ii) High route acquisition latency (iii) Single point of failure (iv) Does not support QoS (v) Non efficient traffic distribution (vi) Frequent link failure
Reactive + Mesh-based + Soft-State	(i) Loop-free (ii) High route acquisition latency (iii) Does not support QoS (iv) Resilient to path failure
Proactive + Hybrid + Hard-State	(i) Low route acquisition latency (ii) Does not support QoS

Discussion. OPHMR is, in the long run, 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, in comparison with other protocols, such as ODMRP [35], while keeping the control packet overhead at an acceptable rate. OPHMR follows the *proactive Hard-State* approach to maintain the multicast topology. Hence, the packet delivery ratio decreases as the mobility of the nodes increases.

5.1.1. Summary of IPLM. Table 2 summarizes the common pros and cons of the IPLM that are in the same subcategory. We have chosen the following subcategories: routing scheme, multicast topology, and maintenance approach, because they have a significant effect on the performance of the multicast routing protocol, on control overhead and packet delivery ratio, for example.

5.2. Application Layer Multicasting (ALM). Overlay multicasting, or ALM, builds a virtual infrastructure to form an overlay network on top of the physical network. Each link in the virtual infrastructure is a unicast tunnel in the physical network. In spite of the advantages of overlay multicasting previously mentioned, it has not been widely deployed [16–18, 25, 36, 53, 68]. This section presents some existing overlay multicast routing protocols. Figure 29 shows an illustration of the ALM multicasting architecture.

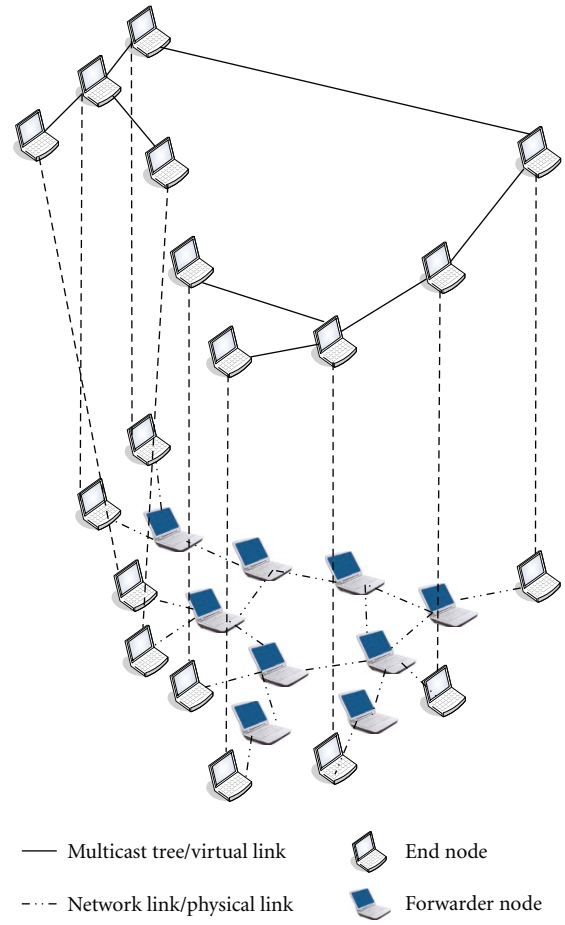


Figure 29: An illustration of application layer multicast.

Ad Hoc Multicasting Routing Protocol (AMRoute)

Protocol Description. AMRoute [16] creates a multicast shared-tree over mesh. It creates a bidirectional shared multicast tree using unicast tunnels to provide connections between multicast group members. Each group has at least one logical core that is responsible for group members and tree maintenance. Initially, each group member declares itself as a core for its own group of size 1. Each core periodically floods *JREQs* (using an ERS) to discover other disjoint mesh segments for the group. Figure 30(a) shows the formation of two disjoint mesh segments (segment 1 and segment 2). When a member node (node Y in Figure 30(b)) receives a *JREQ* from a core (node X in Figure 30(b)) of the same group but a different mesh segment, it replies with a *JACK*, and a new bidirectional tunnel is established between nodes X and Y. Any member, either core or noncore in the mesh segment, can respond to the *JREQ* message to avoid adding many links to a core. Since mesh segments I and II merge, the new mesh contains two logical cores (X and Y). According to the core resolution algorithm, only one of them will be the logical core (say core Y). After the mesh has been created, the logical core periodically transmits *TREECREATE* packets to mesh neighbors in order to build a multicast shared tree. When a member node receives a nonduplicate *TREECREATE*

from one of its mesh links, it forwards the packet to all other mesh links. If a duplicate *TREECREATE* packet is received, a *TREE-CREATE-NAK* is sent back along the incoming link. The node receiving a *TREE-CREATE-NAK* marks the link as a mesh link instead of a tree link. The nodes wishing to leave the group send the *JNAK* message to the neighbors and do not forward any data packets for the group. Figure 30(b) shows the merging of two segments, segment I and segment II, and a virtual user multicast tree.

Discussion. AMRoute creates an efficient and robust shared tree for each group. It helps 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 nonoptimal trees, and requires higher overhead to assign a new core. Also, AMRoute suffers from a single point of failure of the core node.

Progressively Adapted Sub-Tree in Dynamic Mesh (PAST-DM)

Protocol Description. PAST-DM [25] is an overlay multicast routing protocol that creates a virtual mesh spanning all the members of a multicast group. It employs standard unicast routing and forwarding to fulfill multicast functionality. A multicast session begins with the construction of a virtual mesh, on top of the physical links, spanning all group members. Each member node starts a neighbor discovery process using the ERS technique [59]. For this purpose, *Group_REQ* messages are periodically exchanged among all the member nodes. When node X receives a *Group_REQ* message from node Y, it records node Y as its neighbor in the virtual mesh, along with the hop distance to reach node Y. Node X then sends back a *Group_REP* message to Y, so that node Y will record it. The maximum degree of the virtual topology is controlled. The node will stop the neighbor discovery process when the number of virtual neighbors of a node reaches the upper limit. If a node fails to discover any neighbor using the ERS technique, it can use flooding to locate neighbors.

Each source constructs its own data delivery tree based on its local link state table using the source-based Steiner tree algorithm. Let $ds(n)$ denote the hop distance from source node s to node n, then the distance between the source node to a virtual link (n_1, n_2) can be defined as $ds(n_1, n_2) = \min[ds(n_1), ds(n_2)]$. If $c(n_1, n_2)$ denotes the cost of the virtual link (n_1, n_2), then the "*adaptive cost*" of this link to the source is given as $ac(n_1, n_2) = ds(n_1, n_2) * c(n_1, n_2)$. The source can create its Steiner tree by selecting the smallest ac links. Moreover, the source marks all its neighbors as its children in the multicast tree and partitions the remaining nodes into subgroups. Each subgroup forms a subtree rooted at one of the first-level children. The source node does not need to compute the whole multicast tree. It puts each subgroup into a packet header, combines the header with a copy of the data packet, and unicasts the packet to the corresponding children. Each child is responsible for forwarding the data packet to all nodes in its subgroup. It does so by repeating the Source-based Steiner tree algorithm. This process continues until the subgroup is empty or until there is only one member in the subgroup. In the latter case, it unicasts the packet to the receiver. Figure 31 shows an example of this. At the source node S, its receiver list includes R_1, R_2, R_3, R_4, and R_5. Figure 31(a) shows its local view of the virtual topology. The Source-based Steiner tree using adapted costs is shown in Figure 31(b). S generates two smaller lists, R_3 and R_4, R_5. They are included in the header of the packets sent to R_1 and R_2, as shown in Figure 31(c).

When a node intends to join the multicast group, it starts with a normal neighbor discovery process. As the member nodes of the intended group respond with *Group_REP* messages, it can collect its own virtual neighbors and set up its own link state. As the responding group nodes also include the newcomer as their neighbor, they will start to exchange link state tables with the new member. To leave the group, a member node needs to unicast a *Group_LV* message to its current virtual neighbors.

Discussion. PAST-DM constructs a virtual mesh topology, which has the advantage of scaling very well, since this topology can hide the real network topology, regardless of the network dimension. In addition, it uses unicast routing to carry the packets. Moreover, PAST-DM alleviates the redundancy in data delivery in the existence of the change of the underlying topology. However, the link cost calculation may be incorrect, since PAST-DM does not explicitly consider node mobility prediction in the computation of the *adaptive cost*. In addition, the overlay is constructed and maintained even if no source has multicast data to transmit. Exchanging link state information with neighbors and the difficulty of preventing different unicast tunnels from sharing the same physical links may affect the efficiency of the protocol. Simulations [25] show that PAST-DM is more efficient than AMRoute.

Application Layer Multicast Algorithm (ALMA)

Protocol Description. ALMA [18] is an adaptive receiver-driven protocol that constructs an overlay multicast tree of logical links between the group members in a dynamic, decentralized, and incremental way. This approach is based on Round Trip Time (RTT) measurements to detect and manage node mobility. When periodic measurements of the RTT to and from its parent exceed a threshold, a node must perform a reconfiguration procedure on its delivery tree. Each edge of this tree represents a logical link, which corresponds to a path at the network layer. It employs the receiver-driven approach, where each group member finds a parent node on its own, and, once it joins, it can decide to facilitate zero or more children. The parent of a node is the first node on the logical path from the node to the root along the tree. When a node receives a packet from the source, it makes multiple copies of the packet and forwards a copy to each of its children. Members are responsible for maintaining their connections with their parent. If the performance drops

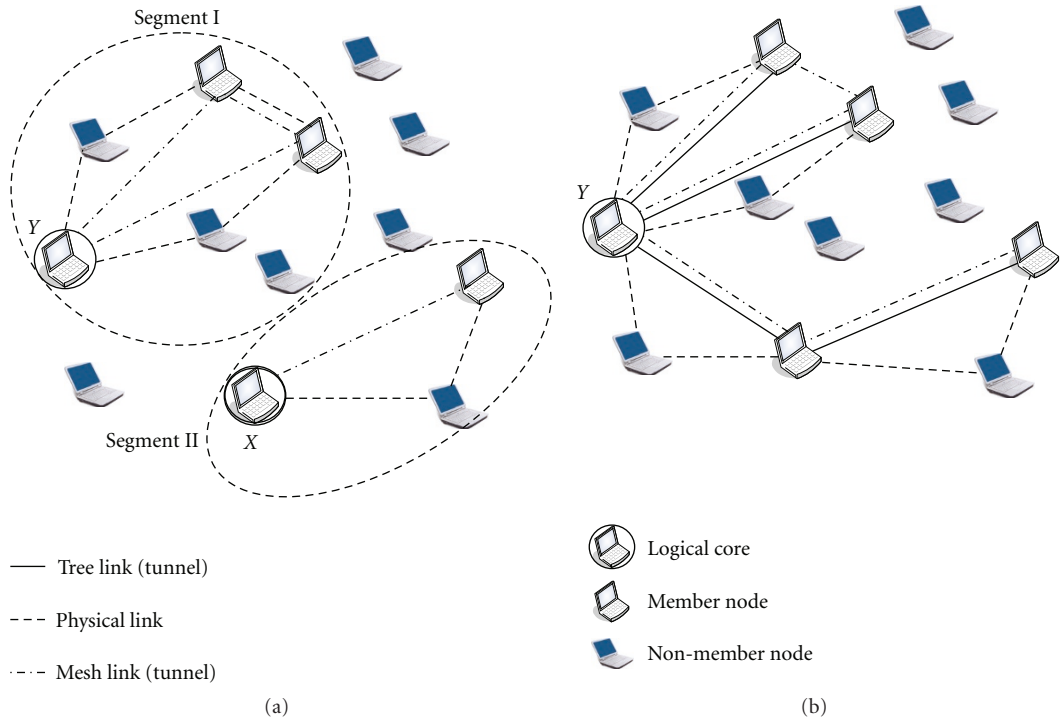


Figure 30: (a) Formation of mesh segments. (b) A virtual user-multicast tree.

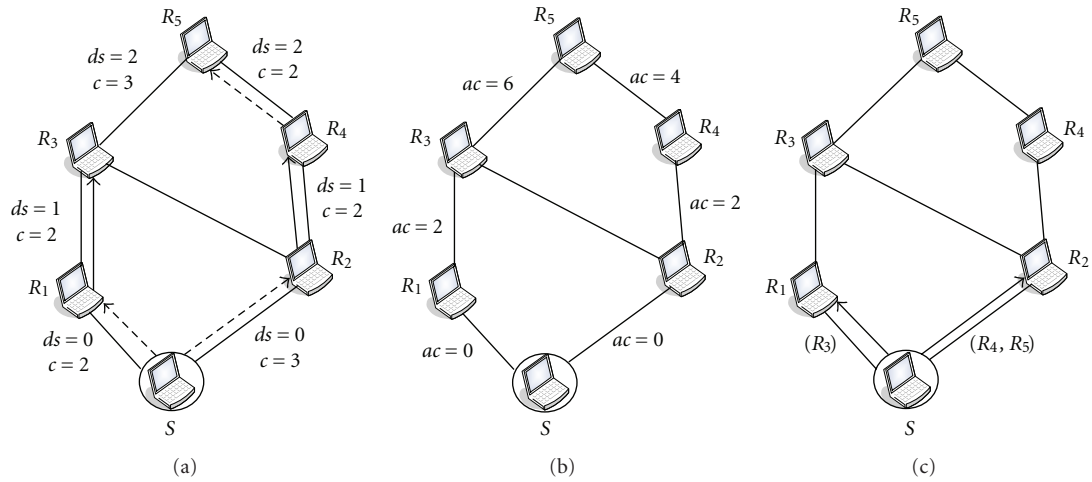


Figure 31: Example of Tree Construction: (a) distance and cost of each link, (b) adapted cost of each link and Source-Based Steiner tree, (c) receiver lists in the header of the packets sent from S to its children. © IEEE 2003.

below a user- or application-defined threshold, the member reconfigures the tree locally, either by switching parents or by releasing children. A new member joins the group by sending join messages, possibly to multiple existing members. An existing member willing to "take" a new child responds to this message. If a new node receives multiple replies, it picks the member whose reply arrives first. When a member wants to leave the group, it is required to send an explicit *Leave* message to both its parent and its children. The parent will delete the node from its list of children, and its children then attempt to rejoin the multicast group.

Discussion. ALMA has the advantages of an application layer protocol, namely, simplicity of deployment, independence from lower-layer protocols, and the capability of exploiting features such as reliability and security which may be provided by the lower layers. However, it employs the ERS [59] to detect neighbors. This makes ALMA, which runs over

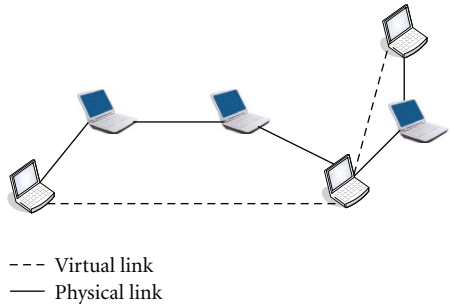


Figure 32: Physical link versus virtual link.

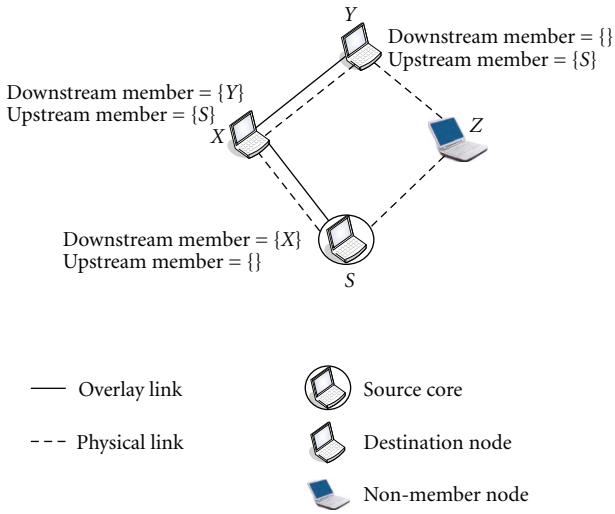


Figure 33: ODOMP overlay example. © IEEE 2005.

costly positioning systems that incur considerable amount of control traffic, more likely to contribute to the overall congestion in the network, although simulations [18] show that ALMA is more efficient than PAST-DM.

On-Demand Overlay Multicast Protocol (ODOMP)

Protocol Description. ODOMP [36] is a reactive protocol which creates an overlay among the group members on demand. The overlay created is a source-rooted tree which connects the group members via IP-in-IP tunnels. When the source node must send a multicast data packet and does not have a valid overlay for this packet, it buffers the packet and initiates the overlay creation by broadcasting a *JREQ* message to its neighbors. When a neighbor node receives a nonduplicate *JREQ*, and, if the node is a group member, it stores the *lastMember* (the address of the last group member that has forwarded this *JREQ*) as its upstream member for this group. It also sets the *lastMember* field of the *JREQ* to its own address and the *distLastMember* field (containing the distance to this member) to zero. After that, it unicasts a *JREP* message to its upstream member and immediately forwards the *JREQ* because the *distLastMember* field is zero. If a nongroup member receives a nonduplicate *JREQ*, the value of the *distLastMember* field is only increased by one, and it waits for (*distLastMember* ∗ *PER_HOP_DELAY*) before it rebroadcasts the *JREQ* to its neighbors. This process continues and eventually a source-rooted tree is created. If a source still has multicast data packets to send, but does not have a valid overlay, it creates a new overlay in the same way. When a group member fails or leaves, a link failure is formed. Such a failure will be corrected during the next recreation of the overlay multicast tree. Figure 33 shows an example of an ODOMP overlay.

Discussion. ODOMP deploys a mechanism called "delayed forwarding," which means that a nongroup member waits for a period of time before rebroadcasting a *JREQ*. The effect of this mechanism is that the *JREQs* of far away group members are suppressed by the "faster" *JREQs* of closer group members. By using the delayed forwarding mechanism, the probability is very high that the *lastMember* in the first *JREQ* received by a node is the closest member of the group. If the *lastMember* is not the closest group member, ODOMP does not fail, but only creates a temporarily less efficient overlay. An effective way to deal with link failure is to deploy the receiver-initiated join mechanism or to send a periodic copy of the *JREP* to the upstream member.

Table 3: Common pros and cons of ALM.

Common pros/cons
(i) Dependent on unicast protocol
(ii) Does not support QoS
(iii) High packet delivery ratio

Table 4: Common pros and cons of MACM.

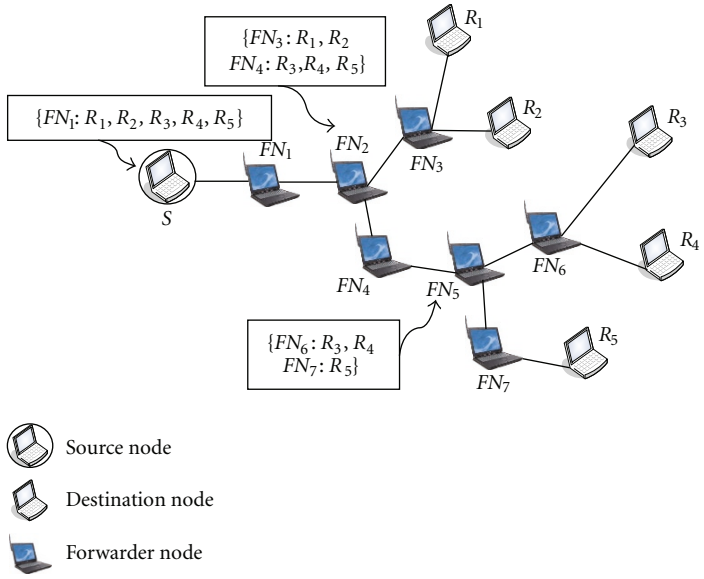
Common pros/cons
(i) High end-to-end delay
(ii) High packet delivery ratio
(iii) Not scalable

5.2.1. Summary of ALM. Table 3 summarizes the common pros and cons of the multicast routing protocols at the application layer.

5.3. MAC Layer Multicasting (MACM). Several schemes have been proposed to provide MAC layer support for multicast communication. These schemes are aimed at providing reliable and efficient multicast at the MAC layer [54, 69–73].

MAC Layer Multicast in Wireless Multihop Networks

Protocol Description. In [54], a MAC protocol which can improve the efficiency of multicast communication is suggested. These authors have developed MAC layer multicast as an extension to the IEEE 802.11 DCF protocol, which can be used with any multicast routing protocol, and introduced several modifications to it to implement their protocol. They modified the control packets (RTS, CTS, and ACK) and



(a)

Xcast header

EMH

Node	Next hop	Destination list
S	FN_1	R_1, R_2, R_3, R_4, R_5
FN_2	FN_3	R_1, R_2
	FN_4	R_3, R_4, R_5
FN_5	FN_6	R_3, R_4
	FN_7	R_5

(b)

Figure 34: Concepts of Xcast and EMH: (a) Xcast packet delivery tree, (b) EMH at node S, FN_2, and FN_5. © IEEE 2004.

data packets. The RTS frame is modified (extended RTS, RTSExt) to include at most four multicast next hop neighbor addresses, a design choice which keeps the RTS frame size within bounds. The CTS frame is modified (extended CTS, CTSExt) to include the order of the receiver (the node that sends the CTS in this case), as determined from the RTS frame, which helps the original sender to differentiate among multiple CTS. The ACK frames are modified (extended ACK, ACKExt) to include the receiver's order determined from the position index (the sending nodes set an integer number in the RTSExt frame, to differentiate among multiple CTSExt, and in the DATA frame, to differentiate among multiple ACKExt) of its address in the received DATA frame. Finally, the DATA packet header is modified (DATAExt) to include the addresses of all those nodes from which CTS was successfully received. Neighbors are grouped into different cliques and multicast data are sent to at most 4 neighbors at a time. Only those nodes that are part of the multicast route and whose addresses are included in the RTSExt must prepare to respond with CTSExt frames. If all the CTSExt frames are sent simultaneously, they may not be correctly received, and so CTSExt are sent one after another by deliberately introducing a fixed amount of delay between successive transmissions.

Discussion. Compared with MMP [72], this protocol has a small RTS size, and so is not prone to collisions. The RTS/CTS/Data/ACK exchange is completed before the NAV of two hop neighbors and potential interferers expire. However, considerable overhead is introduced to transmit a single data packet. In addition, delays may be introduced when paths contain a large number of hops, since each node should wait for some time (calculated as described in [54]) before sending CTSExt and ACKExt. Also, a clustering algorithm is needed when there are more than four next hop nodes in the multicast route.

Batch Mode Multicast MAC/Location Aware Multicast MAC (BMMM/LAMM)

Protocol Description. In [69], two schemes were proposed to provide a reliable MAC layer multicast. The first scheme, known as Batch Mode Multicast MAC (BMMM), uses a similar mechanism of polling to the one used in [74]. However, to avoid the collision of CTS frames that occurred in [74], the transmission of RTS/CTS is in strict sequential order to each of the destinations in BMMW. To prevent collisions among the ACK frames, the transmitter polls each of the neighbors by sending a new packet, called RAK (Request to ACK). This scheme adds considerable overhead to the transmission of a single DATA packet. The second scheme, known as Location Aware Multicast MAC (LAMM), attempts to avoid the control overhead of BMMW by assuming the location information of each of the nodes. It helps the sender to poll only a subset of nodes based on their location.

Discussion. BMMM requires n RAK/ACK exchange pairs and n RTS/CTS exchange pairs for the transmission of a single data packet to n destinations, which means that BMMM is not scalable and is not practical, adding a great deal of control overhead, especially in high traffic networks [75]. In addition, BMMM does not fully utilize the broadcast nature of the broadcast medium, thereby wasting bandwidth. Compared with BMW, BMMM has many fewer contention phases involving the completion of the reliable multicast transmissions. Finally, BMMM will fail in a dense network, because there is contention among many nodes concurrently in the 2-hop neighborhood.

Broadcast Medium Window (BMW)

Protocol Description. A reliable multicast protocol based on round-robin polling is proposed in [70, 73]. Data packets are delivered in straight sequential order. A sending node exchanges RTS/CTS packets with its next-hop neighbors in round-robin order. The RTS packet will carry two additional fields (the multicast session ID and the sequence number of the current packet x). The polled receiving node will respond with the expected sequence number y. Upon receiving the CTS packet, the sending node transmits the packets numbered from y to x. Other receiving nodes that overhear the data packets will buffer the packets and update their expected sequence number. However, this protocol only guarantees that the polled receiving node is free from the interference of hidden terminals.

Discussion. BMW requires that data packets be delivered in straight sequential order, which means that more buffer size is required at the intermediate nodes. In addition, BMW has the following drawbacks [76]. The length of the RTS packet is approximately doubled, which increases the probability of RTS packet collision and communication overhead. In addition, the traffic load is increased substantially if the sending node has a moderate-to-high fanout. Moreover, under high traffic load, the network will be jammed, because of the large amount of retransmission, and the system must fall back on the use of the basic blind broadcast scheme. However, this will defeat the purpose of the reliable multicast protocol, because it cannot be applied when it is most needed. It has been reported in [69] that BMW is inefficient, since it requires at least n contention phases for each multicast data frame. Not only is each contention phase lengthy in terms of time but also the sender must contend with other nodes for access to the medium. It is possible that some other nodes will successfully contend, which will interrupt and prolong the ongoing multicast process. In addition, in many applications (e.g., routing), multicast is time-sensitive. In other words, if the multicast request cannot be fulfilled within a certain amount of time, the multicast request will be considered unsuccessful by the higher layer. For such applications, the prolonged multicast process can easily lead to a time-out in that layer.

A Reliable Multicast MAC Protocol for Wireless Ad Hoc Networks (RMAC)

Protocol Description. In [71], the authors propose Reliable-MAC, a sender-initiated protocol which uses separate channels for the busy tone to achieve reliability for multicast traffic in the MAC layer. Busy tones are used instead of CTS and ACK packets, which reduces the physical layer overhead. Moreover, two busy tones (each with its own narrow bandwidth channel) are introduced, namely, Receiver Busy Tone (RBT) and the Acknowledgment Busy Tone (ABT). RBT is used in the same way as suggested in [77] to eliminate the hidden node problem, and its use is extended to multiple receivers, letting every receiver set up the RBT during data reception. RBT is superior to the RTS/CTS mechanism in addressing the hidden node problem, because, first, data reception is guaranteed to be collision-free, which greatly reduces the number of retransmissions (recall that the RTS/CTS mechanism cannot completely avoid frame collisions), and second, RBT exempts nodes from maintaining the NAV variable needed in the RTS/CTS mechanism, thereby simplifying the protocol. ABT is used to acknowledge the data frames, that is, the receiver will reply with an ABT to the sender if a data frame is correctly received. Using ABT to perform acknowledgments has the following two advantages over using frames: first, an ABT does not need the physical layer preamble and header prepended to a frame, so it can be very short (only long enough to be detected); and second, an ABT does not suffer from collisions or bit errors.

Discussion. Implementation of busy tones in RMAC can, to a large extent, prevent data frame collisions and solve the hidden-terminal problem. However, busy tones require a separate channel, which increases hardware complexity. In addition, RMAC will fail in a dense network, where too many nodes contend simultaneously in the 2-hop neighborhood.

Multicast Aware MAC Protocol (MMP)

Protocol Description. In [72], the authors proposed a multicast aware MAC protocol (MMP) to provide MAC layer support for multicast traffic by attaching an Extended Multicast Header (EMH) containing the information about the next hop that is supposed to receive the multicast packet. The concept of the EMH is similar to that of the Xcast header, the only difference being the inclusion of the IDs of the next hops only. Figure 34 illustrates the difference between the two schemes. The MAC layer then uses the EMH field to support an ACK-based data delivery from the sender to all the receivers on the same multicast subflow. After sending the data packet, the transmitter waits for the ACK from each of its destinations in a strictly sequential order, thereby avoiding the contention between the ACK packets on the sender side. A retransmission of the multicast packet is performed only if the ACK from any of the nodes in the EMH is missing. The retransmission is performed using a technique similar to RTS/CTS, but this time using Multicast RTS or MRTS/CTS.

Discussion. Since there is no upper bound on the number of next hops that can be included in the RTS frame, the RTS packet is larger than the packet size in IEEE 802.11, making the RTS frame itself prone to collisions caused by the problem of hidden terminals. Another disadvantage is that MMP relies on the fact that each next hop receiver is able to correctly receive the CTS frames sent by other receivers.

5.3.1. Summary of MACM. Table 4 summarizes the common pros and cons of the multicast routing protocols at the MAC layer.

Table 5: Comparison of different multicast routing protocols in MANETS.

Layer of operation	Protocol	Routing approach	Unicast routing protocol[(a)]	Loop-free	Route acquisition latency	Control packet flooding	Periodic control message	QoS support	Multicast Control Overhead
Network	BEMRP	Flat	Autonomous	Yes	High	Yes	No	No	Low
	ABAM	Flat	Autonomous	Yes	High	No	No	No	Low
	DDM	Flat	Dependent	Yes	High	Yes	Yes	No	Low
	WBM	Flat	Autonomous	Yes	Low	Yes	No	No	Low
	MZRP	Hierarchy	Unicast-based (ZRP)	Yes	High	Yes	Yes	No	Low
	MCEDAR	Hierarchy	Unicast-based (CEDAR)	Yes	Low	Yes	No	Yes	High
	ITAMAR	Flat	Dependent (any reactive)	Yes	High	No	No	No	High
	PLBM	Flat	Unicast-based (PLBR)	Yes	Low	No	Yes	No	High
	PPMA	Flat	Autonomous	Yes	Low	N/A	N/A	Yes	Low
	ADMR	Flat	Autonomous	Yes	High	No	No	No	Low
	MAODV	Flat	Unicast-based (AODV)	Yes	High	Yes	Yes	No	Low
	AMRIS	Flat	Autonomous	Yes	High	Yes	Yes	No	Low
	MMAs	Hierarchy	Unicast-based (AODV)	Yes	High	No	Yes	No	Low
	ASTM	Hierarchy	Dependent	Yes	Low	Yes	Yes	No	High
	ODMRP	Flat	Autonomous	Yes	High	Yes	Yes	No	Low
	DCMP	Flat	Autonomous	Yes	High	Yes	Yes	No	Low
	FGMP	Flat	Dependent (any reactive)	Yes	High	Yes	Yes	No	Low
	CAMP	Flat	Dependent (any proactive)	Yes	Low	No	Yes	No	High
	NSMP	Flat	Autonomous	Yes	High	Yes	Yes	No	Low
	ACMRP	Flat	Autonomous	Yes	High	Yes	Yes	No	Low
	MANSI	Flat	Autonomous	Yes	High	Yes	Yes	No	Low
	PUMA	Flat	Autonomous	Yes	Low	No	Yes	No	Low
	SRMP	Flat	Unicast-based (DSR)	Yes	High	No	No	Yes	Low
	OGHAM	Hierarchy	Autonomous	Yes	High	No	No	No	Low
	Fireworks	Hierarchy	Autonomous	Yes	High	No	Yes	No	Low
	ABMRS	Hierarchy	Autonomous	Yes	High	No	Yes	No	High
	OPHMR	Flat	Autonomous	Yes	High	No	Yes	No	Low
Application	AMRoute	Flat	Dependent	No	Low	Yes	Yes	No	High
	PAST-DM	Flat	Dependent	Yes	Low	No	Yes	No	High
	ALMA	Flat	Dependent	No	Low	No	Yes	No	High
	ODOMP	Flat	Dependent	Yes	High	Yes	No	No	Low

(a) (i) Unicast-based: means that the multicast protocol depends on a *specific* unicast routing protocol.
(ii) Dependent: the multicast protocol depends on unicast routing protocol.
(iii) Autonomous: the multicast protocol does not depend on unicast routing protocol.

6. Multicast Evaluation Criteria

There are various criteria for evaluating multicast routing protocols, including, but not limited to, packet delivery ratio (PDR), delivery efficiency, protocol efficiency, average latency, number of total packets transmitted per data received, packet retransmission overhead, data forwarding overhead, and control overhead [12, 14, 18, 41, 78–80].

(i) *Packet delivery ratio (PDR):* this expresses the number of non duplicate data packets successfully delivered to each destination versus the number of data packets supposed to be received at each destination. It is a metric which can be used as a measure of the effectiveness of the protocol. The higher the PDR, the more efficient and reliable the protocol.

(ii) *Total overhead:* this represents the total number of control packets transmitted and the total number of data packets transmitted versus the total number of data packets delivered. Total overhead is a more important metric than control overhead because we are concerned about the number of packets transmitted to obtain the number of data packets delivered to the receivers, regardless of whether those packets were data or control. It is a measure which shows efficiency in terms of channel access, and is very important in ad hoc networks, since link layer protocols are typically contention-based. (Control overhead represents the number of control packets transmitted (request, reply, acknowledgment) for each data packet that is successfully delivered to the destinations. Control packets are counted at each hop. This metric can be used as a measure of the effectiveness of a multicast protocol. An ineffective multicast protocol will generate a large number of control packets. It shows, relatively, the degree to which an extra wireless channel access is required for the protocol to exchange control information.),

(iii) *Average latency (average end-to-end latency)*: this represents the average time a data packet takes to travel from the transmitter to the receiver. It is a metric which can be used to evaluate the timeliness of the protocol.

(iv) *The data delivery delay between two nodes*: this represents the average time it takes for a data packet to be transmitted from one forwarding node to another.

(v) *Delivery efficiency*: this represents the number of data packets delivered per data packet transmitted. The term of "transmitted" includes "transmitted by sources" as well as "retransmitted by intermediate nodes." The larger its value, the smaller the number of retransmissions.

(vi) *Reachability (RE)*: this represents the number of all destination nodes receiving the data message divided by the total number of all destination nodes that are reachable, directly or indirectly, from the source nodes.

(vii) *Average throughput*: *receiver throughput* is defined as the total amount of data a receiver R actually receives from all the senders of the multicast group divided by the time it takes for R to receive the last packet. The average over all the receivers is the *average receiver throughput* of the multicast group. The *average throughput* is the average receiver throughput divided by the number of senders.

(viii) *Stress*: the stress of a physical link is the number of identical copies of a multicast packet that needs to traverse the link. This metric quantifies the efficiency of the overlay multicast scheme.

7. Comparison and Summary

Table 5 summarizes the major features of the multicast routing protocols described earlier. It provides a comparison of those protocols in terms of various characteristics: routing approach, dependency on unicast routing protocol, routing scheme, route acquisition latency, and multicast control overhead, and in terms of the presence or absence of the following characteristics: loop-free, control packet flooding periodic control message, QoS support. Concerning the qualitative evaluation in Table 5, it should be noted that most of the values used have relative meaning (comparative evaluation), taking into account that no absolute values are used.

As mentioned previously, the existing multicast routing protocols in MANETs can be classified into various categories. Table 6 lists these various classifications, along with the individual protocols that belong to each subgroup.

8. Conclusion and Future Work

8.1. Conclusion. This paper presented a survey of the existing multicast routing protocols designed for MANETs. We have proposed to classify them into three categories according to their layer of operation, namely, the network layer, the application layer, and the MAC layer, and we also presented various classifications based on different characteristics, namely, multicast topology, initialization of multicast connectivity, routing information update mechanism, and multicast group maintenance. We also described several multicast routing protocols according to the classifications we provided, stating their advantages and drawbacks. The major issues and challenges facing multicast routing design are also presented. These issues should be considered in the design of an efficient multicast routing protocol in MANETs. A multicast protocol can hardly satisfy all previous requirements. In other words, one size does not "fit all," but rather each protocol is designed to provide the maximum possible requirements, according to certain required scenarios. Even if a multicast protocol meeting all the requirements is designed, it will be very complicated and need a tremendous amount of routing information to be maintained. Moreover, it will not be suitable for environments with scarce resources. Satisfying most of the requirements would provide support for reliable communication, minimize storage and resource

Table 6: Classification of multicast routing protocols in MANETS.

Multicast Routing Protocols													
Routing scheme			Initialization of multicast connectivity		Multicast topology				Multicast topology maintenance				
						Tree-based					Hard state		Soft State
Proactive	Reactive	Hybrid	Source initiated	Receiver initiated	Hybrid	Source-Tree-based	Shared-Tree-based	Mesh-based	Stateless	Hybrid	Reactive	Proactive	(Proactive)
ASTM	MAODV	MZRP	AMRIS	PLBM	AMRoute	DDM	AMRIS	MANSI	DDM	AMRoute	CAMP	WBM	DDM
AMRoute	MMAs	OPHMR	NSMP	FGMP	CAMP	ITAMAR	MAODV	DCMP		ASTM	AMRIS		MZRP
CAMP	AMRIS		MZRP	WBM	MCEDAR	ABAM	MMAs	ODMRP		MCEDAR	BEMRP		DCMP
MCEDAR	ODMRP		ODMRP	BEMRP	PAST-DM	PLBM		NSMP		OGHAM	ABAM		FGMP
PAST-DM	FGMP		DCMP	DDM		MCEDAR		FGMP		PAST-DM	PLBM		NSMP
ALMA	ITAMAR		ABAM	MMAs		WBM		ACMRP			SRMP		ODMRP
PLBM	ADMR		ITAMAR	MANSI		MZRP		CAMP			OGHAM		PUMA
	DCMP		PPMA	PUMA		ADMR		PUMA			ADMR		ACMRP
	NSMP		ADMR	MAODV		BEMRP		SRMP			MAODV		ALMA
	ACMRP		ACMRP	SRMP		ALMA		OPHMR			MCEDAR		PAST-DM
	ABAM		OGHAM	ALMA		ODOMP		ABMRS			AMRoute		ODOMP
	SRMP		ODOMP	ASTM		PPMA					ASTM		ITAMAR
	ODOMP		OPHMR	Fireworks		Fireworks					MMAs		MANSI
	BEMRP										ABMRS		Fireworks
	MANSI										OPHMR		
	OGHAM												
	DDM												
	WBM												
	PPMA												
	PUMA												
	Fireworks												
	ABMRS												

consumption, ensure optimal paths (not necessarily as a function of the number of hops), and minimize network load.

8.2. Future work. We believe that research on the use of multicast in mobile ad hoc networks is still in its infancy. Open issues include QoS guarantee, reliable multicast, security provisioning, power efficiency, congestion control, scalability, and efficient membership updates. It is difficult to design a multicast routing protocol that takes all these issues into consideration, that is, a one-size-fits-all design. One possible solution would be to develop an adaptive approach to routing, and this may be the best way forward. Possible topics for future research on multicast routing protocols include the following.

(i) *Interoperability*: most of the existing multicast routing protocols for mobile ad hoc networks were not designed to interoperate with other networks such as wired networks, wireless mesh networks, WiMAX, and so forth. However, it is difficult to design a multicast routing protocol that performs efficiently in a mobile ad hoc network while still being able to interoperate with other networks. In order to offer seamless interoperation, novel mechanisms must be developed to achieve the best performance.

(ii) *Interaction*: mobile ad hoc networks would typically have to support different simultaneous network applications, such as unicast and broadcast. The performance of almost all the existing multicast routing protocols is evaluated in isolation of unicast and broadcast protocols. The interaction effects of unicast and/or broadcast on the performance of multicast routing protocols must be investigated, since these interactions may significantly alter the behavior of multicast protocols.

(iii) *Heterogeneity*: the nodes in mobile ad hoc networks are expected to be heterogeneous with a set of multicast destinations that differ greatly in their QoS requirements and end devices. Existing multicast protocols are developed under the assumption that destinations wish to receive all the information sent by a source. Hierarchical encoding techniques [81, 82] are proposed for the efficient use of resources in heterogeneous networks. These techniques are a layered way of encoding information, such as audio and video, to obtain different quality levels. New multicast protocols must be developed for layered multicast. Not all destinations in layered multicast receive the same amount of data. Each destination has its preferred number of audio/video layers according to its QoS requirements. Open issues include multicast tree construction, audio/video layer assignment, and (re)joining/leaving a multicast group.

(iv) *Integration*: in general, we believe that no single multicast routing protocol is optimal for all mobile ad hoc network scenarios, given the diverse nature and wide range of operating conditions. Therefore, it is desirable to design multicast protocols that adapt well to network conditions (mobility, traffic load, etc.) and optimize functional and performance requirements, such as reliability, control overhead, QoS, and security.

(v) *Mobility*: the continuous and random mobility of nodes in mobile ad hoc networks can easily make the information derived from the network topology stale. As a result, group membership information, such as leaving or joining a multicast group, may induce frequent updates on the protocol states. Moreover, the transmission of data packets can be obstructed during the update process. Thus, group membership approaches should efficiently cope with membership changes in order to minimize their impact on the overall performance of the protocol.

(vi) *Congestion control*: adjacent nodes in mobile ad hoc networks compete with each other to access the wireless medium and transmit their packet. Thus, the network can be easily congested. Congestion, especially in dense networks, introduces long end-to-end delay and buffer overflows and decreases reliability. Instead of leaving the MAC layer to deal with congestion control, multicast protocols should deploy additional novel mechanisms to overcome this congestion.

(vii) *Power efficiency*: the nodes in mobile ad hoc networks are battery operated. Thus, the multicast protocols should provide data delivery with the minimum level of power usage. Multicast protocols that optimize the use of battery power in order to maximize the lifetime of the network should be explored.

(viii) *Network Coding*: the notion of performing coding operations on the contents of packets while they are in transit through the network, commonly called network coding, was originally developed for wired networks. However, it can also be applied with success to MANETs. The main advantage of network coding, substantial performance gains, can be seen in multicast scenarios: improvement in multicast capacity and better resource utilization (in terms of energy consumption, e.g.).

Acknowledgment

This research was supported by a grant from NSERC.

References

[1] IETF MANET Working Group, http://www.ietf.org.
[2] C.-K. Toh, *Ad Hoc Mobile Wireless Networks: Protocols and Systems*, Prentice-Hall, Englewood Cliffs, NJ, USA, 2002.
[3] C. Perkins, *Ad-Hoc Networking*, Addison-Wesley, Reading, Mass, USA, 2000.
[4] S. Deering, "Host extensions for IP multicasting," 1989, http://www.ietf.org/rfc/rfc1112.txt.
[5] S. Paul, *Multicasting on the Internet and Its Applications*, Kluwer Academic Publishers, Norwell, Mass, USA, 1998.

[6] I. Stojmenović, Ed., *Handbook of Wireless Networks and Mobile Computing*, John Wiley & Sons, New York, NY, USA, 2002.

[7] C.-C. Chiang, *Wireless network multicasting*, Ph.D. dissertation, Department of Computer Science, University of California, Los Angeles, Calif, USA, 1998.

[8] M. Gerla, C.-C. Chiang, and L. Zhang, "Tree multicast strategies in mobile, multihop wireless networks," *ACM/Baltzer Journal on Mobile Networks and Application*, vol. 4, no. 3, pp. 193–207, 1999.

[9] C.-C. Chiang, M. Gerla, and L. Zhang, "Adaptive shared tree multicast in mobile wireless networks," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '98)*, vol. 3, pp. 1817–1822, 1998.

[10] C.-C. Chiang, M. Gerla, and L. Zhang, "Shared tree wireless network multicast," in *Proceedings of the 6th International Conference on Computer Communications and Networks (ICCCN '97)*, pp. 28–33, 1997.

[11] C.-C. Chiang and M. Gerla, "Routing and multicast in multihop, mobile wireless networks," in *Proceedings of the 6th IEEE International Conference on Universal Personal Communications (ICUPC '97)*, vol. 2, pp. 28–33, San Diego, Calif, USA, October 1997.

[12] C.-C. Shen and C. Jaikaeo, "Ad hoc multicast routing algorithm with swarm intelligence," *Mobile Networks and Applications*, vol. 10, no. 1, pp. 47–59, 2005.

[13] C. W. Wu, Y. C. Tay, and C.-K. Toh, "Ad hoc Multicast Routing protocol utilizing Increasing id-numberS (AMRIS)," draft-ietf-manet-amris-spec-00.txt, 2000.

[14] S. Park and D. Park, "Adaptive core multicast routing protocol," *Wireless Networks*, vol. 10, no. 1, pp. 53–60, 2004.

[15] J. G. Jetcheva and D. B. Johnson, "Adaptive demand-driven multicast routing in multi-hop wireless ad hoc networks," in *Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '01)*, pp. 33–44, 2001.

[16] J. Xie, R. R. Talpade, A. McAuley, and M. Liu, "AMRoute: ad hoc multicast routing protocol," *Mobile Networks and Applications*, vol. 7, no. 6, pp. 429–439, 2002.

[17] J. Biswas and S. K. Nandy, "Application layer multicasting for mobile ad-hoc networks with network layer support," in *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks (LCN '04)*, pp. 24–31, 2004.

[18] M. Ge, S. V. Krishnamurthy, and M. Faloutsos, "Application versus network layer multicasting in ad hoc networks: the ALMA routing protocol," *Ad Hoc Networks*, vol. 4, no. 2, pp. 283–300, 2006.

[19] C.-K. Toh, G. Guichal, and S. Bunchua, "ABAM: on-demand associativity-based multicast routing for ad hoc mobile networks," in *Proceedings of the IEEE Vehicular Technology Conference (VTC '00)*, vol. 3, pp. 987–993, 2000.

[20] T. Ozaki, J. B. Kim, and T. Suda, "Bandwidth-efficient multicast routing for multihop, ad-hoc wireless networks," in *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '01)*, vol. 2, pp. 1182–1191, 2001.

[21] S. Cai, X. Yang, and W. Yao, "The comparison between PoolODMRP and PatchODMRP," in *Proceedings of the IEEE International Conference on Networks (ICON '03)*, pp. 729–735, 2003.

[22] J. J. Garcia-Luna-Aceves and E. L. Madruga, "The core-assisted mesh protocol," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1380–1394, 1999.

[23] L. Ji and M. S. Corson, "Differential destination multicast-a MANET multicast routing protocol for small groups," in *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '01)*, vol. 2, pp. 1192–1201, 2001.

[24] S. K. Das, B. S. Manoj, and C. S. R. Murthy, "A dynamic core based multicast routing protocol for ad hoc wireless networks," in *Proceedings of the International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '02)*, pp. 24–35, 2002.

[25] C. Gui and P. Mohapatra, "Efficient overlay multicast for mobile ad hoc networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '03)*, vol. 2, pp. 1118–1123, 2003.

[26] C.-C. Chiang, M. Gerla, and L. Zhang, "Forwarding Group Multicast Protocol (FGMP) for multihop, mobile wireless networks," *ACM-Baltzer Journal of Cluster Computing*, vol. 1, no. 2, pp. 187–196, 1998.

[27] Sajama and Z. J. Haas, "Independent-tree ad hoc multicast routing (ITAMAR)," *Mobile Networks and Applications*, vol. 8, no. 5, pp. 551–566, 2003.

[28] R. S. Prasun Sinha and V. Bharghavan, "MCEDAR: multicast core-extraction distributed ad hoc routing," in *Proceedings of the Wireless Communications and Networking Conference*, vol. 3, pp. 1313–1317, 1999.

[29] E. M. Royer and C. E. Perkins, "Multicast ad hoc on demand distance vector (MAODV) routing," Internet-Draft, draft-ietf-draft-maodv-00.txt, 2000.

[30] X. Wang, F. Li, S. Ishihara, and T. Mizuno, "A multicast routing algorithm based on mobile multicast agents in ad-hoc networks," *IEICE Transactions on Communications*, vol. E84-B, no. 8, pp. 2087–2094, 2001.

[31] S. Cai, N. Yao, N. Wang, W. Yao, and G. Gu, "Multipath passive data acknowledgement on-demand multicast protocol," *Computer Communications*, vol. 29, no. 11, pp. 2074–2083, 2006.

[32] X. Zhang and L. Jacob, "MZRP: an extension of the zone routing protocol for multicasting in MANETs," *Journal of Information Science and Engineering*, vol. 20, no. 3, pp. 535–551, 2004.

[33] S. Lee and C. Kim, "Neighbor supporting ad hoc multicast routing protocol," in *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '00)*, pp. 37–44, 2000.

[34] C.-C. Hu, E. H.-K. Wu, and G.-H. Chen, "OGHAM: on-demand global hosts for mobile ad-hoc multicast services," *Ad Hoc Networks*, vol. 4, no. 6, pp. 709–723, 2006.

[35] S.-J. Lee, W. Su, and M. Gerla, "On-demand multicast routing protocol in multihop wireless mobile networks," *Mobile Networks and Applications*, vol. 7, no. 6, pp. 441–453, 2002.

[36] O. Stanze and M. Zitterbart, "On-demand overlay multicast in mobile ad hoc networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '05)*, vol. 4, pp. 2155–2161, 2005.

[37] M. Lee and Y. K. Kim, "PatchODMRP: an ad hoc multicast routing protocol," in *Proceedings of the International Conference on Information Networking*, pp. 537–543, 2001.

[38] S. Cai and X. Yang, "The performance of poolODMRP protocol," in *Proceedings of the IFIP/IEEE International Conference on Management of Multimedia Networks and Services (MMNS '03)*, vol. 2839, pp. 90–101, 2003.

[39] R. S. Sisodia, I. Karthigeyan, B. S. Manoj, and C. S. R. Murthy, "A preferred link based multicast protocol for wireless mobile ad hoc networks," in *Proceedings of the IEEE International Conference on Communications (ICC '03)*, vol. 3, pp. 2213–2217, 2003.

[40] R. Vaishampayan and J. J. Garcia-Luna-Aceves, "Protocol for unified multicasting through announcements (PUMA)," in *Proceedings of the IEEE International Conference on Mobile Ad-Hoc and Sensor Systems (MASS '04)*, 2004.

[41] U. T. Nguyen and X. Xiong, "Rate-adaptive multicast in mobile ad-hoc networks," in *Proceedings of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob '05)*, vol. 3, pp. 352–360, 2005.

[42] H. Moustafa and H. Labiod, "SRMP: a mesh-based protocol for multicast communication in ad hoc networks," in *Proceedings of the International Conference on Third Generation Wireless and Beyond 3Gwireless*, pp. 43–48, May 2002.

[43] S. K. Das, B. S. Manoj, and C. S. R. Murthy, "Weight based multicast routing protocol for ad hoc wireless networks," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '02)*, vol. 1, pp. 117–121, 2002.

[44] D. Pompili and M. Vittucci, "PPMA, a probabilistic predictive multicast algorithm for ad hoc networks," *Ad Hoc Networks*, vol. 4, no. 6, pp. 724–748, 2006.

[45] C. S. R. Murthy and B. S. Manoj, *Ad Hoc Wireless Networks: Architectures and Protocols*, Prentice-Hall, Upper Saddle River, NJ, USA, 2004.

[46] M. Ilyas, Ed., *Handbook of Ad Hoc Wireless Networks*, CRC Press, Boca Raton, Fla, USA, 2002.

[47] Z. Wang, Y. Liang, and L. Wang, "Multicast in mobile ad hoc networks," in *Proceedings of the IFIP International Federation for Information Processing*, vol. 258, pp. 151–164, 2008.

[48] S. Papavassiliou and B. An, "Supporting multicasting in mobile ad-hoc wireless networks: issues, challenges, and current protocols," *Wireless Communications and Mobile Computing*, vol. 2, no. 2, pp. 115–130, 2002.

[49] X. Chen and J. Wu, "Multicasting techniques in mobile ad hoc networks," in *The Handbook of Ad Hoc Wireless Networks*, pp. 25–40, 2003.

[50] L. Junhai, X. Liu, and Y. Danxia, "Research on multicast routing protocols for mobile ad-hoc networks," *Computer Networks*, vol. 52, no. 5, pp. 988–997, 2008.

[51] K. Chen and K. Nahrstedt, "Effective location-guided tree construction algorithms for small group multicast in MANET," in *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '02)*, vol. 3, pp. 1180–1189, 2002.

[52] C. K. Wong, M. Gouda, and S. S. Lam, "Secure group communications using key graphs," *IEEE/ACM Transactions on Networking*, vol. 8, no. 1, pp. 16–30, 2000.

[53] K.-I. Kim and S.-H. Kim, "DESIRE: density aware heterogenous overlay multicast forwarding scheme in mobile ad hoc networks," *IEICE Transactions on Communications*, vol. E88-B, no. 9, pp. 3579–3586, 2005.

[54] S. Jain and S. R. Das, "MAC layer multicast in wireless multihop networks," in *Proceedings of the 1st International Conference on Communication System Software and Middleware (Comsware '06)*, pp. 1–10, 2006.

[55] C. Gui and P. Mohapatra, "Scalable multicasting in mobile ad hoc networks," in *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, vol. 3, pp. 2119–2129, 2004.

[56] Z. J. Haas and M. R. Pearlman, "The Zone Routing Protocol (ZRP) for Ad Hoc Networks," draft-zone-routing-protocol-00.txt, 1997.

[57] R. Sivakumar, P. Sinha, and V. Bharghavan, "CEDAR: a core-extraction distributed ad hoc routing algorithm," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1454–1465, 1999.

[58] R. S. Sisodia, B. S. Manoj, and C. S. R. Murthy, "A preferred link based routing protocol for wireless ad hoc networks," *Journal of Communications and Networks*, vol. 4, no. 1, pp. 14–21, 2002.

[59] C. E. Perkins, "Ad Hoc On Demand Distance Vector (AODV) Routing," Internet-Draft, draft-ietf-manet-aodv-00.txt, 1997.

[60] S. Y. Oh, J.-S. Park, and M. Gerla, "E-ODMRP: enhanced ODMRP with motion adaptive refresh," *Journal of Parallel and Distributed Computing*, vol. 68, no. 8, pp. 1044–1053, 2008.

[61] X. Xiong, U. T. Nguyen, and H. L. Nguyen, "Preemptive multicast routing in mobile ad-hoc networks," in *Proceedings of the International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies (ICN/ICONS/MCL'06)*, pp. 68–74, 2006.

[62] T. Ballardie, P. Francis, and J. Crowcroft, "Core based trees (CBT)," *ACM SIGCOMM Computer Communication Review*, vol. 23, pp. 85–95, 1993.

[63] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*, T. Imielinski and H. Korth, Eds., vol. 5, pp. 153–181, 1996.

[64] L. K. Law, S. V. Krishnamurthy, and M. Faloutsos, "A novel adaptive protocol for lightweight efficient multicasting in ad hoc networks," *Computer Networks*, vol. 51, no. 3, pp. 823–834, 2007.

[65] S. S. Manvi and M. S. Kakkasageri, "Multicast routing in mobile ad hoc networks by using a multiagent system," *Information Sciences*, vol. 178, no. 6, pp. 1611–1628, 2008.

[66] A. B. Mnaouer, L. Chen, C. H. Foh, and J. W. Tantra, "OPHMR: an optimized polymorphic hybrid multicast routing protocol for MANET," *IEEE Transactions on Mobile Computing*, vol. 6, no. 5, pp. 503–514, 2007.

[67] P. Jacquet, P. Minet, A. Laouiti, L. Viennot, T. Clausen, and C. Adjih, "Multicast optimized link state routing," Internet Draft, draft-ietf-manet-olsr-molsr-01.txt, 2002.

[68] K.-D. Kim, J.-H. Park, K. Lee, H.-Y. Kim, and S.-H. Kim, "A scalable location-based application layer multicast protocol in manet," in *Proceedings of the IASTED International Conference on Wireless Networks and Emerging Technologies, Part of the 6th IASTED International Multi-Conference on Wireless and Optical Communications*, 2006.

[69] M.-T. Sun, L. Huang, S. Wang, A. Arora, and T.-H. Lai, "Reliable MAC layer multicast in IEEE 802.11 wireless networks," *Wireless Communications and Mobile Computing*, vol. 3, no. 4, pp. 439–453, 2003.

[70] K. Tang and M. Gerla, "MAC reliable broadcast in ad hoc networks," in *Proceedings of the IEEE Military Communications Conference (MILCOM '01)*, vol. 2, pp. 1008–1013, 2001.

[71] W. Si and C. Li, "RMAC: a reliable multicast MAC protocol for wireless ad hoc networks," in *Proceedings of the International Conference on Parallel Processing (ICPP '04)*, pp. 494–501, 2004.

[72] H. Gossain, N. Nandiraju, K. Anand, and D. P. Agrawal, "Supporting MAC layer multicast in IEEE 802.11 based MANETs: issues and solutions," in *Proceedings of the Conference on Local Computer Networks (LCN '04)*, pp. 172–179, 2004.

[73] K. Tang and M. Gerla, "Congestion control multicast in wireless ad hoc networks," *Computer Communications*, vol. 26, no. 3, pp. 278–288, 2003.

[74] K. Tang and M. Gerla, "Random access MAC for efficient broadcast support in ad hoc networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '00)*, vol. 1, pp. 454–459, 2000.

[75] A. Chen, D. Lee, G. Chandrasekaran, and P. Sinha, "HIMAC: high throughput MAC layer multicasting in wireless networks," in *Proceedings of the IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS '07)*, pp. 41–50, 2007.

[76] K. S. Lau and D. Pao, "Multicast medium access control in wireless ad hoc network," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '04)*, vol. 3, pp. 1903–1908, 2004.

[77] F. A. Tobagi and L. Kleinrock, "Packet switching in radio channels—part II: the hidden terminal problem in carrier sense multiple-access and the busy-tone solution," *IEEE Transactions on Communications*, vol. 23, no. 12, pp. 1417–1433, 1975.

[78] M. S. Corson and J. Macker, "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations," Internet Engineering Task Force, 1999.

[79] S.-J. Lee, W. Su, J. Hsu, M. Gerla, and R. Bagrodia, "Performance comparison study of ad hoc wireless multicast protocols," in *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '00)*, vol. 2, pp. 565–574, 2000.

[80] Y.-S. Chen, T.-S. Chen, and C.-J. Huang, "SOM: spiral-fat-tree-based on-demand multicast protocol in a wireless ad-hoc network," *Computer Communications*, vol. 25, no. 17, pp. 1684–1695, 2002.

[81] N. Shacham, "Multipoint communication by hierarchically encoded data," in *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '92)*, vol. 3, pp. 2107–2114, 1992.

[82] V. K. Goyal, "Multiple description coding: compression meets the network," *IEEE Signal Processing Magazine*, vol. 18, no. 5, pp. 74–93, 2001.