

# CHAPTER 2

---

## Wireless Mesh Communication Architectures and Protocols

---

### 2.1 Introduction

A wireless mobile mesh network (WM<sup>2</sup>Net) is a network of wireless, possibly mobile, devices that can freely and dynamically self-organize in arbitrary and temporary network topologies. A WM<sup>2</sup>Net may be left unattended after deployment for days, months or even years in areas without any preexisting communication infrastructure. Various sources of failure (e.g., malfunction of nodes and mobility) produce a time-varying communication infrastructure. Communication protocols tailored for operation in a WM<sup>2</sup>Net need thus be designed to operate in such dynamic contexts and cope with this frequently changing multihop network topology.

The limitations on power consumption imposed by portable wireless radios, coupled with the fact that the communication infrastructure does not rely on the assistance of centralized stations, implies that terminals must communicate with each other either directly or indirectly using multihop routing techniques. Unlike traditional wireless networks, WM<sup>2</sup>Nets do not rely on predeployed infrastructure; instead, each individual WM<sup>2</sup>Net node becomes part of the overall infrastructure. In this context, not all WM<sup>2</sup>Net nodes need communicate directly with a high-power control tower or base station or access point, but only with their local peers, instead. Peer-to-peer networking protocols provide a mesh-like interconnect to shuttle data between the thousands of tiny devices in a multihop fashion.

With these unique characteristics in mind coupled with the fact that WM<sup>2</sup>Net nodes are small sized objects with built in nonreplaceable batteries, a networking mechanism to be applicable in a WM<sup>2</sup>Net must be simple, energy efficient, scalable, and robust. Hence, implementation of traditional networking protocols such as routing and medium access control (MAC), with no concern about energy consumption, scalability, and fault tolerance, is not practical in these networks.

Examining the structure and operation of WM<sup>2</sup>Net networks can offer insights into ways in which communications could be supported within massively populated wireless mesh configurations. Given that WM<sup>2</sup>Net nodes are devices with very limited memory, communication bandwidth, and processing resources, in order to support large-scale

## 18 Chapter Two

network deployments, anywhere from hundreds to millions of mesh nodes, efficient and lightweight WM<sup>2</sup>Net protocols are needed. The design of communication protocols tailored for operation in such dynamic contexts turns out to be a very challenging engineering goal.

---

### 2.2 WM<sup>2</sup>Net Configurations

A WM<sup>2</sup>Net configuration can be either *hierarchical*, *flat* or *hybrid*. In a *hierarchical* or *infrastructure* network, nodes are partitioned into groups, often called *clusters*. Generally, there are three kinds of nodes in a cluster, namely, the cluster-head node, the gateway (GW) node, and the cluster-member (internal) node. Cluster-head (CH) nodes basically emulate the functionalities of an AP. All nodes in a cluster can communicate with their CH and (possibly) with each other (of the same cluster) (Iwata et al., 1999).

Various different heuristics can be used for the CH election. These may include node addresses, node degrees (neighbor connectivity), transmission power and mobility, or more sophisticated node weights combining the above attributes (see Chatterjee et al., 2002; Singh and Raghavendra, 1998).

GW nodes are used to provide connectivity among clusters. To communicate within a cluster, a GW must select the frequency or code used by that cluster. GW nodes can communicate with multiple CHs and thus can be reached via multiple paths. Consequently, similar to a router in the wireline Internet, which is equipped with multiple subnet addresses, a GW may have multiple hierarchical addresses.

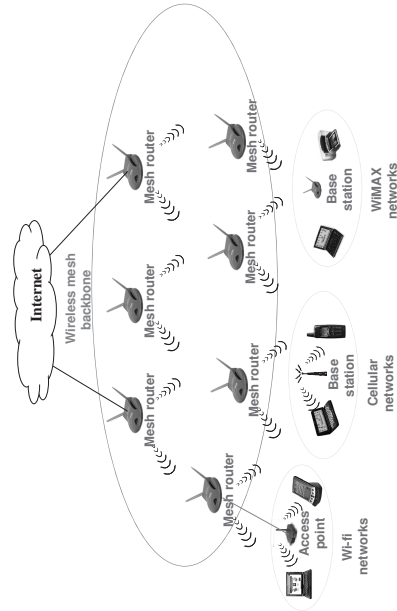
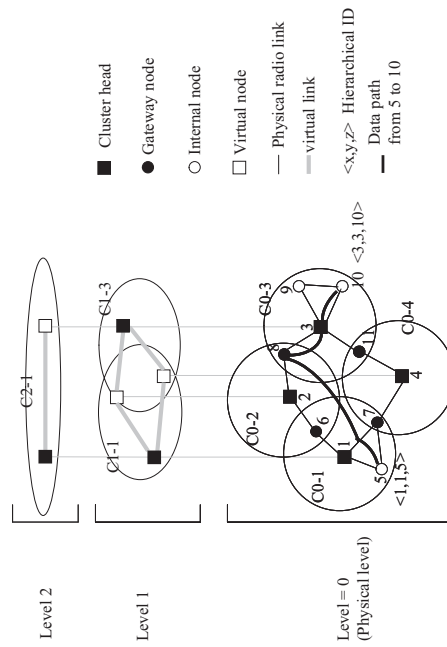
Depending on the number of hierarchies (levels), the depth of the network can vary from a single tier to multiple tiers. Figure 2.1 illustrates a two-tier example. At Level = 0, there appear 4 physical-level clusters C0-1, C0-2, C0-3, and C0-4. Level 1 and level 2 clusters are generated by recursively selecting CHs (Iwata et al., 1999).

For a node A in a cluster X (Fig. 2.2), to establish communication with some node B at some different cluster, say Y, its traffic must first be routed to its CH. From the CH, traffic is then routed to a GW node, to another CH, and so on until the CH of the destination node (cluster Y) is reached. Traffic is then delivered to the destination node.

Furthermore, infrastructured WM<sup>2</sup>Net architectures enable the integration with existing wireless networks (see Fig. 2.1) through GW/bridge functionalities built in mesh routers.

In a flat, or client, WM<sup>2</sup>Net architecture there is no grouping and all nodes have equal responsibilities. Connections are established between nodes that are in close enough proximity to allow sufficient radio propagation conditions to establish connectivity. In this form of architecture, client nodes constitute the actual network to perform routing and configuration functionalities as well as to provide customers with end-user applications. Clients thus function both as mesh router for routing and self-configuration, and as end user. A packet destined to a node in the network hops through multiple nodes to reach the destination. An example of a flat network is depicted in Fig. 2.3.

The *hybrid* WM<sup>2</sup>Net architecture is the combination of infrastructure and client meshing as shown in Fig. 2.4. Mesh clients can access the network through mesh routers as well as through meshing with other mesh clients. While the infrastructure provides connectivity to other networks such as the Internet, Wi-Fi, WiMAX (see Section 5.4, for details), cellular, and sensor networks, the routing capabilities of clients provide improved connectivity and coverage inside the WM<sup>2</sup>Net.



**FIGURE 2.1** An example of physical/virtual clustering.

20 Chapter Two

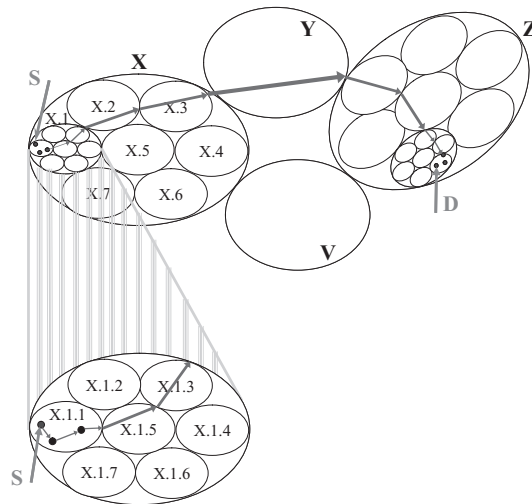


FIGURE 2.2 Hierarchical routing.

### 2.3 Hierarchical versus Flat WM<sup>2</sup>Net Architecture

The major advantage of the *hierarchical* architecture is the ease of the mobility and resource management process. Mobility of nodes within an infrastructure-less mobile wireless network raises organizational problems quite different and rather more challenging than those for wireless communication (cellular) networks (Mouly and Pautet, 1992). As there is no centralized administrative control, rapid response to nodal movement requires *adaptive, autonomous, and distributed* organization mechanisms that involve minimal manual intervention.

The aggregation of nodes into clusters provides a convenient framework for the development of important features, including channel reuse among clusters (in terms of frequency, time, or spreading code) (Gilhousen et al., 1991), channel access and channel and bandwidth allocation (Gerla and Tsai, 1995; Gilhousen et al., 1991). In cluster-based

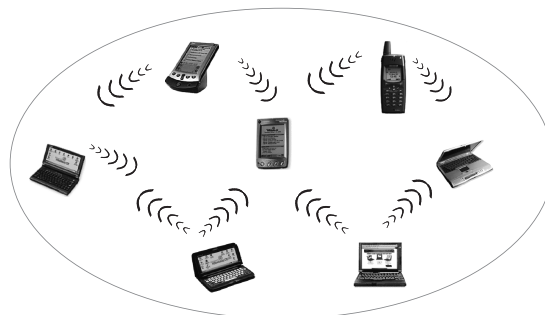


FIGURE 2.3 Flat or client WM<sup>2</sup>Net.

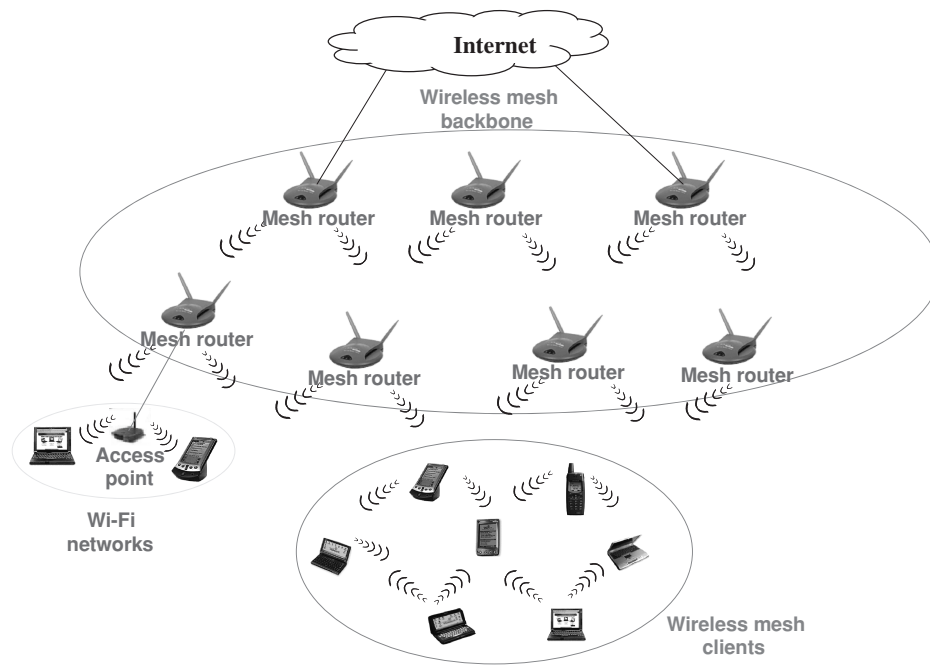


FIGURE 2.4 Hybrid WM<sup>2</sup>Nets.

networks, a CH node acts as a local coordinator within its cluster: it keeps track of the member nodes of its cluster (network management), accounts for resources so that bandwidth reservations can be placed on them (in a deterministic or statistical sense), and assists in locating nodes outside its cluster during the course of a data transmission (Gerla and Tsai, 1995). This approach is in line to that followed in cellular communication networks, where such resource accountability is facilitated by the fact that all stations learn of each other's requirements through a control station (e.g., base station in cellular systems). In addition, to reduce the overhead inherent to routing and processing (Iwata et al., 1999), complete routing information is maintained only for intracluster routing (Lin and Gerla, 1997), whereas for intercluster routing the topology details are hidden through hierarchical aggregation techniques. For networks composed of a large number of mobile devices, a hierarchical network configuration would be a practical and scalable solution (Klein et al., 1997; Xu et al., 1998).

Furthermore, clustering can provide a nice methodology for accomplishing quality-of-service (QoS) guarantees in a mobile network. Ensuring QoS communications in a wireless mesh network is apparently highly dependent upon routing and resource management control: conforming to QoS measures, such as delay bounds, depends on the quality of the chosen route whereas, in addition, complying with QoS guarantees imposes the use of a MAC method that guarantees the successful transmission of packets under high mobility and/or heavy load circumstances. This can hardly be made feasible without a central regulatory authority to carry out the significant functions of routing and channel (resource) management in WM<sup>2</sup>Nets.

## 22 Chapter Two

The major advantage of clustering stems from the fact that some information about the state of the network is kept local (Broch et al., 1998; McDonald and Znati, 1999). Even in very dynamic situations, not all changes need be propagated throughout the network. Specifically, by confining location update propagation to the lowest level (in the hierarchy) containing the moving endpoint, costs can be made proportional to the distance moved. As with most large organizational problems, specialized node roles and regional node addressing help hierarchical routing protocols to scale with network size, especially when there is a structure in the underlying network connectivity (e.g., group mobility) that can be exploited (McDonald and Znati, 2000).

Notably, mobility and traffic have a great impact on cluster size and design. In large-scale WM<sup>2</sup>Nets, local conditions can vary significantly, so in some regions a larger cluster size might be advisable. However, there is not a single or default parameter, or a set of parameters, such as cluster size and cluster merge/split threshold, that can be fitted for all real networks. Besides, it is unlikely that a single default parameter would be acceptable for all clustered WM<sup>2</sup>Nets. In these terms, a single initial value would be a good starting point and during operation, different heuristics are used to locally optimize cluster parameters.

There are also several features in cluster-based wireless networks, which are potentially complex to implement (Iwata et al., 1999). First, cluster IDs are dynamically assigned. This assignment must be unique—not an easy task in dynamic contexts, where the hierarchical topology is continuously changing. Second, each cluster can dynamically merge and split, based on the number of nodes in the cluster. This feature causes frequent changes of CH, thus degrading the network performance significantly. Since the diameter of a cluster is variable, it is also difficult to predict the time it takes to propagate clustering control messages among nodes. As a consequence, it is difficult to bound the convergence time of the clustering algorithm. Third, the paging and query/response approach used to locate mobile nodes may lead to a nonnegligible amount of control message overhead. Fourth, if the CH leaves its current cluster, this function migrates to another location manager. This requires a complex consistency management between original and new cluster.

Finally, the determination of the CHs shall be done in such a way that the reconfigurations of the network topology are minimized. This is an important issue, since an essential criterion in cluster-based algorithms is *cluster stability*. Frequent CH changes may adversely affect the performance of other functions such as scheduling and resource allocation, which rely on it.

Under the “extreme” scenario where mobility rates are high and mobility patterns random, such that all nodes in the field are moving very rapidly in different random directions, each cluster stays intact for only a very short amount of time and, under this scenario, it seems that clusters would need to be constantly created/modified, thus rendering a lot of cluster maintenance overhead. Therefore, the control signaling generated from the cluster maintenance becomes the bulk of overhead.

In *flat* architectures, each node maintains a routing table with entries for all nodes in the network (Iwata et al., 1999). Flat networks require only one “type” of equipment, as all nodes have to perform the same operation. That is, all nodes are treated as network members, similar to Cluster-Member nodes in hierarchical clustered architectures, with all having the same responsibilities. In addition, nodes in flat networks transmit at a significantly lower power than the transmission power of a CH, which is reasonably higher in order to cover its cluster territory. Operating a network at low power levels

has several implications: first, the battery power of the nodes in WM<sup>2</sup>Nets is preserved. Second, the wireless spectrum can be better reused, leading to more network capacity. Third, and possibly most importantly, a larger degree of low probability of interception and detection can be achieved, resulting in a more secure network operation.

On the other hand, a flat architecture is acceptable if the user population is small. As the number of mobile hosts increases, however, so does the overhead, thus creating scalability concerns when applied to large networks.

To conclude, there are surely circumstances under which a flat WM<sup>2</sup>Net architecture is preferable to a hierarchical one, but one always has the size factor backwards. Hierarchical clustering tends to localize the impact of state changes, so it tends to be more useful in larger networks than in smaller ones. If each node is one or two hops away from every other node, clustering may not be the optimal solution in terms of signaling overhead generated. However, if the network diameter is 10 or 20 hops, a methodology is needed to reduce the size of the routing problem, and clustering seems to be an effective way to achieve this.

---

## 2.4 Routing in Mobile Wireless Networks

In WM<sup>2</sup>Nets, where communication terminals are mobile and the transmission medium is wireless, routing is a major problem. The limitations on power consumption imposed by portable wireless radios, coupled with the fact that the communication infrastructure does not rely on the assistance of centralized stations, imply that terminals must communicate with each other either directly or indirectly using multihop routing techniques. As nodes move about, this results in a distributed multihop wireless network with a time-varying topology.

Before delving into the details of the properties underlying dynamic routing in wireless networks, our primary issue is to find out whether a conventional routing protocol, like link-state or distance-vector, could apply in a wireless multihop environment. To respond, we need first to list several outstanding structural differences that exist between wireline and wireless mobile networks and make routing very different in the two environments (Johnson, 1994; Perkins, 2000; Toh, 2002; Prakash, 1998):

- In a mobile wireless network, the **rate of topological changes** is relatively very high compared to that of wireline networks. As is the case in wireline networks, the procedures for route selection and traffic forwarding in wireless mobile networks require accurate information about the current state of the network (e.g., node interconnectivity and link quality) and the session (e.g., traffic rate, end-point locations), in order to direct traffic along paths that are consistent with the services requirements of the session and the service restrictions of the network.

However, changes in network or traffic sessions are likely to occur more frequently in mobile wireless networks than in stationary wireline networks. The degree of dynamism in route selection depends on several factors, including the type and frequency of changes in network and session state; the limitations on response delay imposed in assembling, propagating, and acting upon this state information; the amount of network resources available for these functions; and the expected performance degradation resulting from a mismatch between selected routes and the actual network and session state.

## 24 Chapter Two

The routing mechanism must be able to quickly detect and respond to such state changes in order to minimize service degradation of existing traffic sessions whereas, at the same time, the algorithm must do so using a minimal amount of network resources, in order to maximize the overall network performance (Ramanathan and Steenstrup, 1996).

On one hand, the effectiveness of a routing protocol increases as network topology information becomes more detailed and up to date. To maintain up-to-date routing tables, a conventional routing protocol should be forced to continuously send and receive topology updates. In WM<sup>2</sup>Nets, however, the topology may change quite often, requiring *frequent exchanges of control information* (e.g., routes, route updates, or routing tables) among the network nodes. In an event-triggered Link-State protocol (Aggélou, 2004) any topological change would trigger a flooding, resulting in a flooding rate equal to the topological change rate. In this scenario, a blind route update mechanism could unnecessarily waste network resources since updates are sent even when no data transmission at all occurs in the network. In addition, as the number of network nodes can be large, the potential number of destinations is also large, thus requiring *a high volume of control information* exchanged among the network nodes. As a consequence, the amount of update traffic can be even higher, the distribution of which can eventually saturate the network.

Notably, radio spectrum is a scarce resource, which means that packet-radio networks typically have limited bandwidth available. Because the wireless devices must share access to the radio channel, the bandwidth available to any node is even more limited. Relatively low bandwidth combined with the potential for routing algorithms to generate large numbers of packets means that efficiency is paramount in designing packet-radio routing algorithms. This observation is, however, in contradiction with the fact that all updates in the wireless communication environment travel over the air and are then costly in resources. An even more disappointing fact is that as the network size increases and as the nodal mobility increases, smaller and smaller fraction of this total amount of control traffic will be even used. This is so, since the more mobile nodes become, the shorter the residual lifetime of a link turns out to be. Thus, the period in which the routing information remains valid decreases as well. Since the rate of link failure is directly related to node mobility, greater mobility increases both the volume of control traffic required to maintain routes and the congestion due to traffic backlogs. Thus, a crucial algorithm design objective in order to achieve routing responsiveness and efficiency is the minimization of reaction to mobility and of exchange of information.

On the other hand, when the rate of topological changes is extremely high, little can be done to ensure that routing algorithms converge fast enough to track topological changes (Iwata et al., 1999; Corson and Ephremides, 1995). In this situation, flooding-based routing algorithms may be the only viable routing option. In this regard, we should also note that under extreme conditions, where the changes in network topology occur too frequently, finding a loop-free path may become impossible too. We conclude then that the topology changes shall occur sufficiently slowly in order to allow successful propagation of topology updates.

- **Broadcast transmissions are unreliable.** Since broadcast packets are not receiver directed, there is no way to reserve the wireless medium at the receivers before transmitting a broadcast packet (e.g., with the use of an RTS/CTS exchange—see Section 5.3.1). Consequently, broadcast packets are inherently less reliable than unicast packets.

This difference does not exist in wireline networks, and presents a fundamental limitation of wireless networks that must be accounted for in the design of WM<sup>2</sup>Net routing protocols. Broch et al. (1998) demonstrate that over any single hop, 99.8% of *unicast data* packets are received successfully, while only 92.6% of *broadcast packets* are received. The difference between the two numbers is attributed to collisions.

- **Wireless links can be asymmetric and unidirectional** (Haas and Tabrizi, 1998; Chambers, 2002; Prakash, 1998). A link between two nodes  $i$  and  $j$  is called unidirectional when node  $i$  can properly receive traffic from a node  $j$  (in this sense,  $i$  can receive data from  $j$  above a certain BER threshold and thus properly decode it), but  $j$  cannot receive properly traffic from  $i$ . As a consequence, a transmission that requires a handshake between  $i$  and  $j$  fails.

A link between two nodes  $i$  to  $j$  is called asymmetric, when the transmission quality of the link (e.g., data rate) from  $i$  to  $j$  is different from that of the link from  $j$  to  $i$ .

- Wireless mesh devices present **technological limitations** on the use of resources, namely, battery power, transmission bandwidth, and CPU time, compared to their wireline counterparts.

With these constraints in mind, routing protocols designed for wireline networks cannot directly apply in WM<sup>2</sup>Nets. In fact, conventional routing protocols would perform very badly (Krishna et al., 1997; Barret et al., 2001; Chambers, 2002), both from a practical standpoint of building such a network, and from a theoretical standpoint in terms of what there seems to be promising routing algorithms, if used in a dynamic environment.

Link-state and distance-vector would probably work very well in a WM<sup>2</sup>Net with low mobility, that is, a network where the topology is not changing very often. The main problem with link-state and distance-vector is that they are designed for a static topology, which means that they would have problems to converge to a steady state in a WM<sup>2</sup>Net with a frequently changing topology. In addition, the problem that still remains that the link-state and distance-vector are highly dependent on periodic control messages. As the number of network nodes can be large, the potential number of destinations is also large. This requires large and frequent exchange of data among the network nodes. This is in contradiction with the fact that all updates in a wireless interconnected mesh network are transmitted over the air and thus are costly in resources such as bandwidth, battery power and CPU.

Based upon these considerations, the desirable qualitative properties of a wireless mesh networking protocol are: (1) to cope with frequently changing network infrastructures; (2) to ensure small convergence time, based on high collaboration among nodes; (3) to be robust given a common spectrum of WM<sup>2</sup>Net conditions, such as high channel congestion and frequently changing topologies; (4) to scale well in large node populations, in terms of storage, computational and transmission overhead; and (5) to be simple.

## 26 Chapter Two

Besides, other more evident desirable qualitative properties include:

- **Scalability:** In its general view, scalability of a networking protocol is its ability to support the continuous increase of the network parameters (as for example traffic rate, network size, etc.) without degrading network performance (Iwata et al., 1999; Santivanez et al., 2002). Notably, one shall distinguish, however, between network scalability and routing protocol scalability. In its general context, network scalability is what the network can support, whereas routing protocol scalability is what the routing protocol can handle provided that the network can. Simply speaking, if the network can support thousands of nodes for a given traffic load, then for a routing protocol to be considered scalable, it should not break when run over that network of thousands of nodes with that traffic load. So, basically, routing protocol scalability means matching (or improving) the network scalability properties.

From (Iwata et al., 1999), it is argued that the routing protocol scalability is dependent on the scalability properties of the network the protocol runs over. That is, the network's own scalability properties provide the reference level as to what to expect from a routing protocol. Obviously, if the overhead induced by a routing protocol grows faster than the network rate but slower than the minimum traffic load, the routing protocol is not degrading network performance. The latter is, in fact determined by the minimum traffic load.

Furthermore, scalability of a routing protocol does not solely depend on its performance (e.g., packet delivery ratio) versus network density, or traffic load, or control overhead, or some combination of performance measures. While we may assume that a protocol A that uses control signaling (control packets) more efficiently (number of packets delivered per control packet) than a protocol B, or, else, both protocols deliver the same number of packets, but one must work harder to do so, can we say that protocol A is more scalable than protocol B, given that the performance of both protocols is the same (at least from an external view)? Measuring the protocol's control overhead does not necessarily provide enough information to extrapolate the results to what will happen when network parameters (size, mobility, etc.) are increased. This is so, since there are other factors, as route suboptimality for instance, that may become more relevant as traffic and network size increase. To this end, a routing protocol that produces less control overhead may be forming longer paths, which may not be an issue at your current traffic rate, but as the traffic rate increases the extra hops may be comparable to (or greater than) the control overhead.

- **Distributed operation:** The protocol should not be dependent on a centralized controlling node.
- **Demand-based operation:** To minimize the control overhead in the network and thus not wasting network resources more than necessary, the protocol should be reactive. This means that the protocol should only react when needed and that the protocol should not periodically broadcast control information.
- **Multiple routes information:** To reduce the number of reactions to topological changes and congestion, multiple routes could be used. If one route has become invalid, it is possible that another stored route could still be valid and thus saving the routing protocol from searching for a new route.

- **QoS support:** The general goal of efficient routing is to get packets reliably from the source to the destination while maximizing the capacity of the network and minimizing the delivery delay. Optimal routing algorithms that maximize capacity or minimize delay typically need an estimate of the network flows, network topology, information about the residual capacity of links, and so on. Some sort of QoS support is then necessary to incorporate into the routing protocol.
- **Power conservation and Sleep period operation:** WM<sup>2</sup>Net nodes can be laptops and thin clients, such as PDAs, as well as micro/nanodevices such as microelectromechanical systems (MEMS) and nanoelectromechanical systems (NEMS) with very limited battery power resources. It is therefore important for the network protocols to support sleep and temporarily inactive modes.
- **Network partition support:** Mobility of nodes together with wireless links of varying quality could lead to overly frequent topology changes. These could further cause to some network segments to become completely disconnected from other network segments.

---

## 2.5 Routing Protocol Categories

Routing has traditionally used the knowledge of the instantaneous connectivity of the network with emphasis on the state of network links. This is the so-called *topology-based* approach (Mauve et al., 2001). An alternative to the topology-based approach, called *location-based (or position-based)* routing, uses information related to the physical position of nodes to help the task of routing (Basagni et al. (1998, 1998a); Bose et al., 1999; Mauve et al., 2001; Capkun et al., 2001; Xue and Li, 2001; De Couto and Morris, 1998; Haas and Liang, 1999; Karp, 2001; Karp and Kung, 2000; Ko and Vaidya (1998, 1999); Li et al., 2000).

An alternative to these approaches, called *power/energy-aware routing*, uses information related to the remaining battery lifetime of mobiles with the goal to produce paths that comprise nodes with a high value of remaining lifetime as well as to help them adjust their transmission power so that to keep the energy required to complete the routing task at minimum levels. Power/energy-aware routing is covered in Chapter 3.

This sequel highlights the specifics of each category, exposing their advantages as well as potential limitations. To develop an intuitive feel of the concepts behind each category, we tabulate a few typical routing techniques per category.

### 2.5.1 Topology-Based Routing Protocols

In the topology-based approach, the associated routing protocols can be classified into the following three general categories, based on the timing when the routes are discovered and updated: *proactive* (also called *table-driven*), *reactive* (also called *on-demand*), and *hybrid*.

### 2.5.2 Proactive (Table-Driven) Routing

The proactive approach is similar to the connectionless approach of traditional datagram networks. In proactive schemes, nodes, based on a periodic update process (Royer and Toh, 1999; Bruce McDonald and Taieb Znati, 1999), attempt to compute a priori and maintain consistent, up-to-date routing information to all nodes in the network, regardless of whether the routes are being used for carrying packets.

## 28 Chapter Two

A routing protocol is then proactive in the sense that nodes calculate all possible paths to all destinations independently of their effective use such that when a packet needs to be forwarded, the route is already known and can be immediately used.

As is the case for wireline networks, each node maintains a routing table. Routing tables typically contain a list of addresses for all possible destinations, next-hop nodes, and the number of hops to reach each destination node. The routing table is constructed using either link-state or distance-vector algorithms.

### 2.5.2.1 Properties

The main advantage of proactive routing protocols is that, when an application needs to initiate a data call, routing information is immediately available thus eliminating route acquisition delays. In fact, this can be useful in various cases, as in interactive applications.

Although this approach can ensure high quality routes in a static topology, as of a wireline network, it does not scale well to large, highly dynamic networks. In fact, proactive protocols require each node to maintain a large table to store routing information for the entire network. The constant propagation of routing incurs substantial signaling traffic and power consumption. Given that both bandwidth and battery power are scarce resources in mobile devices, pure proactive schemes may not be the appropriate solution for a mobile wireless environment with a large number of nodes (Royer and Toh, 1999; Pearlman and Haas, 1999). A more disappointing observation, however, is that the overhead expended to establish and/or maintain a route between a source-destination pair is wasted if the data source never requires a data path.

Also, since proactive schemes rely on periodic broadcasts, they need some time to converge before a route can be used. This convergence time is probably negligible in a static wireline network, where the topology is not changing so frequently. In a mobile wireless network, on the other hand, where the topology is expected to be very dynamic, this convergence time will probably mean a lot of dropped packets before a valid route is detected.

### 2.5.3 Reactive (On-Demand) Routing

The philosophy behind on-demand routing protocols is to evaluate the network on an as-needed basis and create routes only when there is a need for carrying data traffic. If no data traffic is generated, then the routing activity shall be totally absent. Based on the assumption that not all the routes are used at the same time, a need for route triggers thus a route search in a reactive routing strategy.

Reactive protocols are characterized by the elimination of the conventional routing tables at nodes, and consequently the need of routing table updates to track changes in the network topology. As a result, an on-demand process for discovering routes is a prerequisite; a path discovery is triggered asynchronously when there is a need for data packet and no path to the intended node is known.

The discovery procedure is often based upon a query-reply cycle: the data source node floods the network with a query packet to discover a route to the data destination. Assuming no network disconnections, the destination will be eventually reached by the query. Upon receiving the query, the destination sends a reply back to the source. Since multiple copies of the same query may arrive at the destination via alternate paths, the destination may send more than one reply back to the source, each producing a different route. The source selects then the optimum route, using its own route selection criteria.

The process is completed once a route is found or all possible route permutations have been examined. Once established, the data source uses the route to send its data packets to the destination.

During the data-forwarding phase, routing information is maintained by a maintenance procedure until either the destination becomes inaccessible along all paths from the source or until the route is no longer desired (Royer and Toh, 1999). Should at any time during the data exchange the route maintenance indicates that the route is broken a new route discovery cycle is triggered to set up a new route.

When the communication ends, the source does not attempt to further maintain the connectivity to the destination. Hence, network resources are not expended to maintain routes that are not actively used for data traffic.

### 2.5.3.1 Properties

On-demand routing strategies create and maintain routes between a pair of source-destination only when necessary. Therefore, in contrast to the proactive approach, in reactive protocols the control overhead as well as routing table storage is drastically reduced, when traffic is sparse (Aggélou, 2004 (Chapter 3); Broch et al., 1998; Das et al., 1998; Das et al., 2000; Das et al., 2000a; Jacquet and Viennot, 2000; Johansson et al., 1999; Maltz et al., 1999; Johnson and Maltz, 1996). On-demand routing does scale well thus to large populations, as it does not maintain a permanent routing entry to each destination.

However, similar to connection-oriented communications, a route is not available when needed but upon completion of the Route Discovery phase. This may introduce an initial route setup latency. This latency may in fact be detrimental to certain applications such as interactive applications (e.g., distributed database queries). Moreover, the quality of the data path (e.g., bandwidth, delay etc.) is not known prior to call setup. It can be discovered only while setting up the path, and must be monitored by all intermediate nodes during the session, thus paying the related latency and overhead penalty. Such a priori knowledge is, however, desirable in multimedia applications for call acceptance control decisions as well as bandwidth negotiations (Iwata et al., 1999).

Because of the long route setup delays as well as the lack of path quality information prior to call set-up, pure reactive routing protocols may not be applicable to real-time communications (Giordano, 2000). Table 2.1 lists some of the basic differences between the two classes of algorithms.

### 2.5.4 Hybrid Routing (Haas, 1997; Ramanathan and Steenstrup, 1998; Krishna et al., 1997; Lin and Gerla, 1997; Gerla and Tsai, 1995)

Mobility of nodes in infrastructure-less wireless networks raises organizational problems quite different and rather more challenging than those for infrastructured wireless networks. Mobile wireless networks differ in the frequency and degree at which the topology changes. A protocol that works well in one WM<sup>2</sup>Net may not work well in another with a different density, size, etc. The diverse applications of WM<sup>2</sup>Nets pose, however, a challenge for a single protocol that operates efficiently across a wide range of operational conditions and network configurations. Purely proactive or purely reactive protocols perform well in a limited region of this range. For example, reactive routing protocols are well suited for networks where the “call to mobility” ratio is relatively low. Proactive routing protocols, on the other hand, are well suited for networks where this

## 30 Chapter Two

Parameters	On-Demand	Table-Driven
Availability of routing Information	Available when needed	Always available regardless of need
Routing philosophy	Flat	Mostly flat except for cluster switch gateway routing (CSGR)
Periodic route mobility	Not required	Yes
Coping with mobility	Using localized route discovery and in ABR and SSR	Inform other nodes to achieve consistent routing table
Signaling traffic generated	Grows with increasing mobility of active routes (as in ABR)	Greater than that of on-demand routing
QoS support	Few can support QoS	Mainly shortest path as QoS metric

**TABLE 2.1** Comparisons of On-Demand Versus Table-Driven Routing Protocols

ratio is relatively high. The performance of either class of protocols degrades when the protocols are applied to regions of WM<sup>2</sup>Net space between the two extremes.

Regardless of what type of routing protocol is preferred there will be a set of circumstances under which it will not perform well. Consequently, despite being designed for the same type of underlying network, it does not seem that a routing strategy based exclusively on proactive or reactive routing can achieve the objectives required for WM<sup>2</sup>Net routing (Royer and Toh, 1999). A desirable design objective for an architectural framework capable of supporting routing in large WM<sup>2</sup>Nets subject to high rates of node mobility shall balance the tradeoff between reactive and proactive routing while minimizing the shortcomings of each (Bruce McDonald and Taieb Znati, 1999). So, what is ideally needed is a single routing protocol that has the intelligence to adjust its behavior *dynamically* based on the rate of changes (mobility) and the activity (rate of data) as to match the specific mobility/activity ratio.

Researchers advocate that the issue of “efficient operation over a wide range of conditions” can be addressed by a *hybrid* routing approach, where the “proactive” and “reactive” behavior is mixed in the amounts that best match these operational conditions. Given multiple protocols, each suited for a different region of the WM<sup>2</sup>Net design space, it does make sense to capitalize on each protocol’s strengths by combining them into a single framework (that is, hybridization). In the most basic hybrid framework, one of the protocols would be selected based on its suitability for the specific network’s characteristics. Although not an elegant solution, such a framework would perform as well as the best-suited protocol for any scenario and outperform either protocol over the entire WM<sup>2</sup>Net design space. However, by not using both protocols together, this approach fails to capitalize on the potential synergy that would make the framework perform as well or better than either protocol for any given scenario.

A more promising approach for protocol hybridization is to have the base protocols operate simultaneously, but with different “scopes.” For the case of a two-protocol framework, protocol A could operate locally, while the operation of protocol B would be global. The key to this framework is that the local information acquired by protocol A is

used by protocol B to operate in a more efficient manner. This framework can be tuned to network behavior simply by adjusting the size of the protocol A's scope. In one extreme configuration, the scope of protocol A is reduced to nothing, leaving protocol B to run by itself. As the scope of protocol A is increased, the information provided to protocol B increases as well, thereby decreasing protocol B's overhead. At the other extreme, protocol A is made global, eliminating the load of protocol B altogether. So, at either extreme, the framework defaults to the operation of an individual protocol. In the wide range of intermediate configurations, the framework performs better than either protocol on its own.

It is worth remarking that routing protocols exhibit, to some extent, some degree of multiscope behavior. Certain proactive routing protocols for instance monitor the status of neighbor connectivity through broadcast beacons, which occur at a faster rate than the global Link-State (or Distance-Vector) advertisements.

To highlight all these issues in practical terms, this sequel describes a new dynamic clustering scheme for coverage-time optimization (DC-CTO) in two-tier WM<sup>2</sup>Nets. The coverage-time is defined as the time elapsed until the first CH runs out of power. In regulating cluster regions, DC-CTO scheme achieves balanced power consumption among CHs such that energy-rich CHs progressively increase their cluster region to enlarge their coverage area with higher populations of member nodes.

#### 2.5.4.1 Coverage-Time Optimized Dynamic Clustering for Two-Tiered WM<sup>2</sup>Nets<sup>1</sup>

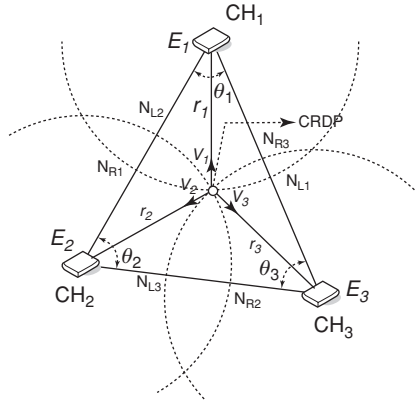
**Dynamic Clustering for DC-CTO Scheme** DC-CTO aims to minimize the energy consumption in CHs, while the entire mesh network remains fully connected. To achieve energy efficiency, cluster ranges are adjustable. Energy-efficient radii are calculated based on the results of DC-CTO. DC-CTO consists of three phases: the initial phase, the dynamic cluster control (DCC) phase, and the transmission power allocation (TPA) phase. The following assumptions are made:

- The architecture of WM<sup>2</sup>Net has a two-tiered hierarchical structure.
- The upper layer comprises the CHs and the lower layer comprises the SNs.
- A sink node is aware of the position of all CHs.

**Initial Phase** In the initial phase, CHs are deployed randomly to construct a triangle that determines "cluster radius decision points (CRDPs)" as shown in Fig. 2.5. The distance between CRDP and each CH is assumed to be equal to the radius of each cluster. To construct a triangle Delaunay triangulation (de Berg et al., 2000; Aurenhammer, 1991) is used, as it guarantees the construction of equilateral triangles. The construction of equilateral triangles leads to balanced energy consumption of each CH.

<sup>1</sup>Excerpt from the invited article "Coverage-time optimized dynamic clustering for two-tiered wireless mesh networks," <sup>†</sup>Joongheon Kim, <sup>‡</sup>Wonjun Lee, <sup>§</sup>Eunkyo Kim, and <sup>¶</sup>Timothy K. Shih (\*This work was jointly supported by grants from the Korea Science and Engineering Foundation (KOSEF) [R01-2005-000-10267-0] and SK Telecom [KU-R040572]; <sup>†</sup>Research engineer, Digital Media Research Lab., LG Electronics, Seoul, Korea; <sup>‡</sup>Faculty member of the Department of Computer Science and Engineering, Korea University, Seoul, Korea, E-mail: wlee@korea.ac.kr); <sup>§</sup>Research engineer, LG Electronics Institute of Technology, LG Electronics, Seoul, Korea; <sup>¶</sup>Faculty member of the Department of Computer Science and Information Engineering Tamkang University, Taiwan).

## 32 Chapter Two



**FIGURE 2.5** System model for a DC-CTO scheme.

**Dynamic Cluster Control (DCC) Phase** Upon completion of the initial phase, DC-CTO goes to the “dynamic cluster control (DCC) phase.” The cluster radii of the three CHs can be dynamically controlled using a CRDP as a pivot. The CH and CRDP distance becomes the radius of each cluster. In this context, the synergy of DC-CTO with DCC can achieve balanced energy consumption among CHs.

In addition to balancing energy consumption among CHs, another goal of the DCC phase is to determine the positions of CRDPs that contribute to the minimum energy consumption in CHs. To achieve this goal, energy-efficient cluster radii are needed. As shown in Fig. 2.5, the triangle is composed of three CHs. If the size of a cluster increases, its CH will consume more energy attributed to the increased population. Therefore, if the overlapping areas of each sector are larger than the optimal, CHs consume more energy than required. It is important thus to find the fewest overlapping areas while full network connectivity is ensured. A CRDP is determined using an energy-constrained objective function that is based on nonlinear programming (NLP) methods with iteration policy. The objective function is illustrated in Eq. (2.1).

$$\min : f = \frac{3}{2} \sum_{i=1}^3 \theta_i r_i^2 \frac{E_i}{\sum_{j=1}^3 E_j} - S$$

$$d_i^2 = (x_{\text{CRDP}} - x_i)^2 + (y_{\text{CRDP}} - y_i)^2 \quad (2.1)$$

where  $S$  denotes the area of the triangle and  $E_i$  denotes the amount of remaining energy at each  $\text{CH}_i$ .

To minimize coverage overlapping, the objective function accounts for the energy of each CH that composes the Delaunay triangle. As an NLP method for solving Eq. (2.1), a “limited memory BFGS (L-BFGS) method” is used; this is a very efficient NLP method for solving the unconstrained optimization problem. Note that all but the angular values can be transmitted to the sink. CHs should, however, know these, which are computed at the sink node using the second law of cosine. The sink node eventually obtains the energy state and position of each CH.

**Algorithm 1** Pseudo Code of DC-CTO scheme

---

```

1: loop
2:   //Expire timer: t
3:   Initialize(t); //t=0;
4:   t- -; //Reducing the expire time
5:   if (event == IDENTIFIED) then
6:     //If event is occurred in the WSNs
7:     numEvt = numEvt + 1; //numEvt: recognition counter
8:     if (numEvt > Threshold) then
9:       //Threshold: threshold for number of identification
10:      Operate DC-CTO scheme: NLP-based approach
11:      Initialize(numEvt); //numEvt=0;
12:      Function Call: Initial Phase
13:      Function Call: DCC Phase
14:      Function Call: TPA Phase
15:    end if
16:    if (t==0) then
17:      //When expire timer is expired
18:      Operate DC-CTO scheme: NLP-based approach
19:      Initialize(numEvt); //numEvt=0;
20:      Function Call: Initial Phase
21:      Function Call: DCC Phase
22:      Function Call: TPA Phase
23:    end if
24:  end if
25: end loop

```

---

**FIGURE 2.6** The pseudo code of a DC-CTO scheme.

**TPA Phase** Having calculated the distance between a point and three neighbor CHs, we can now calculate the optimal assigned transmission power of each CH by using EIRP formula. The pseudo-code of a DC-CTO scheme is illustrated in Fig. 2.6.

### 2.5.5 Position or Location-Aided Routing Protocols (Iwata et al., 1999; Mauve et al., 2001; Jain et al., 2001; Karp and Kung, 2000; Lin and Wang, 1999; Stojmenovic and Lin, 2001)

Position- or geo-based routing protocols use the geographic position of nodes to make routing decisions. Geo-routing is commonly structured around two core functions: the *location estimation function* and the *geographic forwarding function* (Mauve et al., 2001). The former is used to determine the position of the intended destination (sink) node whereas the latter, based on the calculated location information, routes the call to its intended destination.

The specifics of location service and geographic forwarding are discussed in Section 2.5.5.1 and 2.5.5.3, respectively.

#### 2.5.5.1 Methodologies for Location Estimation in WM<sup>2</sup>Nets

Localization is the process of identifying the location of nodes. There are several ways to localize mesh nodes. Nodes can be localized at deployment time, for instance, using a

## 34 Chapter Two

global positioning system (GPS) receiver that is attached to the object or person deploying the nodes. With mesh devices needed to last for months or even years without battery replacement, traditional GPS-based localization techniques are, however, not suited for these requirements. Running GPS ([www.navcen.uscg.gov/gps](http://www.navcen.uscg.gov/gps)) on each device is costly and energy prohibitive for a number of applications, not sufficiently robust to jamming for military applications, and limited to outdoor applications. Besides, the receivers at the lowest end give poor accuracy, with inaccuracies of tens of meters possible (Garmin's eTrex, see at [www.garmin.com/products/etrex/spec.html](http://www.garmin.com/products/etrex/spec.html)). Receivers suitable for operation in WM<sup>2</sup>Net must have an accuracy of submeter. These, however, come with a price of more than \$5,000.

The problem can be remedied, however, by using non-GPS techniques as proposed by Capkun et al. (2001). A survey-grade device can be used to localize nodes after deployment (Girod et al., 2006). A closed-loop system that is equipped with a pan/tilt laser can provide similar accuracy without human intervention (Stoleru et al., 2005; Romer, 2003). Besides, coarse locations can be obtained by simply placing beacon nodes with known positions throughout the deployment area. Nodes can thus estimate their positions based on the beacons within their radio range. Each of these localization techniques achieves a different balance of human effort prior to deployment, node effort after deployment, and localization accuracy.

The following paragraphs discuss techniques for localization using the received signal from one or more reference nodes.

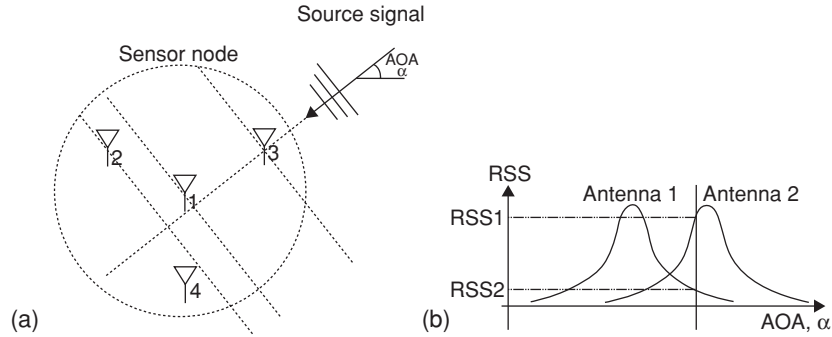
**Received Signal Strength** It is commonly known that the signal strength of a radio message decreases as the distance from the transmitter increases. The received signal strength (RSS) can be converted to a distance estimate given that there exists a mapping from these RSS indicator (RSSI) values to distances. For RF systems (Bahl and Padmanabhan, 2000; Hightower et al., 2000), problems such as multipath fading, background interference, and irregular signal propagation characteristics make range estimates, however, inaccurate. It remains thus questionable whether the distance can be accurately determined based on signal strength, propagation patterns, and fading models.

On the other hand, RSSI is the cheapest and simplest option available to measure distance as RSSI values come for free with all radio devices.

**Time-of-Arrival** Time-of-arrival (ToA) is the reception time of a signal (RF, acoustic, or other); that is, the time of transmission plus a propagation-induced time delay. The cornerstone of time-based techniques is the receiver's ability to accurately estimate the arrival time of the line-of-sight (LoS) signal. This estimation is though hampered both by additive noise and multipath signals.

**Angle-of-Arrival** Angle-of-arrival (AoA) provides information about the direction to neighboring nodes rather than the distance to neighboring nodes. AoA is calculated using RSS and ToA-based methods; thus, it provides complementary information to the ToA and RSS measurements.

There are two common methods that nodes measure AoA (see Fig. 2.7). The first is to use a node array and employ the so-called array signal processing techniques at mesh nodes. In this case, each node embeds a four-element Y-shaped microphone array, as shown in Fig. 2.7a. The AoA is estimated from the differences of arrival times for a transmitted signal in each array element. The estimation is similar to time-delay estimation, as discussed above, but generalized to the case of more than two array elements. When



**FIGURE 2.7** AoA estimation methods. (a) AoA is estimated from the ToA differences among sensor elements embedded in the node; a 4-element Y-shaped array is shown. (b) AoA can also be estimated from the RSS ratio  $RSS1/RSS2$  between directional antennas.

the impinging signal is narrowband (i.e., its bandwidth is much less than its center frequency), then a time delay  $\tau$  relates to a phase delay  $\varphi$  with  $\varphi = 2\pi f_c \tau$  where  $f_c$  is the center frequency. Narrowband AoA estimators are often formulated based on the phase delay.

The second approach to AoA estimation takes advantage of the RSS ratio between two (or more) directional antennas mounted on mesh node (see Fig. 2.7b). Two directional antennas pointed in different directions, such that their main beams overlap, can be used to estimate the AoA using the ratio of their individual RSS values.

Both approaches require the use of multiple antenna elements. This requirement drives up the mesh device cost and size. The reader is referred to the work by Van Veen and Buckley (1988), Stoica and Moses (1997), Ottersten et al. (1993) for further discussions on AoA estimation algorithms and their properties.

**Lateration** In practice, it may not be feasible for all nodes to be equipped with special purpose location determination hardware, but rather a small fraction of the population. Such nodes, called “anchor nodes,” can act as reference points for location information. Mesh nodes use the information from anchor nodes to estimate their own position. A common technique for locating objects using other objects whose position is known is called *lateration*.

In lateration, each node possesses information about the estimated distances ( $d_i$ ) along with the locations ( $x_i, y_i$ ) of a series of anchors. This produces the following set of equations (with  $(x, y)$  being the location of the node, which is initially unknown):

$$\begin{aligned} (x_1 - x)^2 + (y_1 - y)^2 &= d_1^2 \\ &\vdots \\ (x_n - x)^2 + (y_n - y)^2 &= d_n^2 \end{aligned} \quad (2.2)$$

This can be linearized by subtracting the last equation from all others, which results in:

$$\begin{aligned} x_1^2 - x_n^2 - 2(x_1 - x_n)x - y_1^2 - y_n^2 - 2(y_1 - y_n)y &= d_1^2 - d_n^2, \\ &\vdots \\ x_{n-1}^2 - x_n^2 - 2(x_{n-1} - x_n)x - y_{n-1}^2 - y_n^2 - 2(y_{n-1} - y_n)y &= d_{n-1}^2 - d_n^2, \end{aligned} \quad (2.3)$$

## 36 Chapter Two

This can be reordered to a standard system of linear equations:  $Ax = b$ :

$$A = \begin{bmatrix} 2(x_1 - x_n) & 2(y_1 - y_n) \\ \vdots & \vdots \\ 2(x_{n-1} - x_n) & 2(y_{n-1} - y_n) \end{bmatrix}$$

$$b = \begin{bmatrix} x_1^2 - x_n^2 + y_1^2 - y_n^2 + d_n^2 - d_1^2 \\ \vdots \\ x_{n-1}^2 - x_n^2 + y_{n-1}^2 - y_n^2 + d_n^2 - d_{n-1}^2 \end{bmatrix} \quad (2.4)$$

This system can then be solved with standard least-squares calculations:

$$\hat{x} = (A^T A)^{-1} A^T b \quad (2.5)$$

The location estimate  $\hat{x}$  can be verified by calculating the total residue of the given distances ( $d_i$ ) and the distances to location estimate  $\hat{x}$ :

$$residue = \frac{\sum_{i=1}^n \sqrt{(x_i - \hat{x})^2 + (y_i - \hat{y})^2} - d_i}{n} \quad (2.6)$$

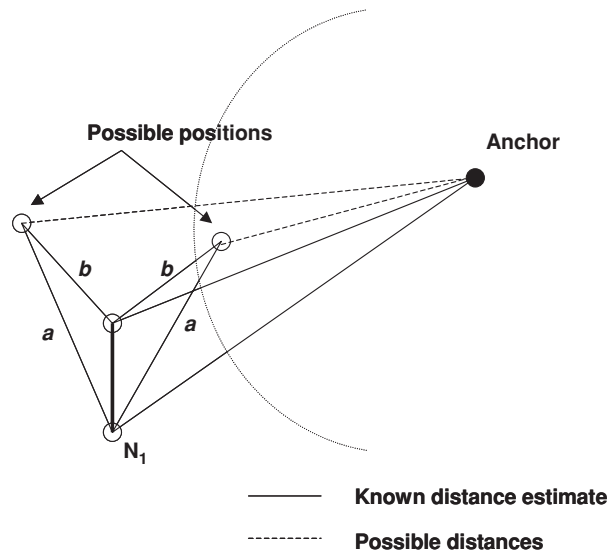
This residue should be as small as possible and at least smaller than the known radio range.

**Min-Max** As shown above, lateration requires quite a lot of calculations. Mesh nodes, however, are relatively limited in processing power, as they do not feature a mathematical coprocessor. Min-max is a simpler approach; it does not work with circles but with square bounding boxes around each anchor with the length of a side being twice the distance estimate to that anchor. A node can now combine two bounding boxes of two anchors; the intersection of those two produces its own location.

This process continues for all known anchors and finally the position for the node is calculated as the center of the resulting intersection. All calculations required are only a few simple additions, subtractions, minimum or maximum. Simplicity makes Min-max a very interesting option for running on mesh nodes.

**Ad Hoc Positioning** Niculescu and Nath (2001) present a new ad hoc positioning method, called the ad hoc positioning system (APS) that extends the capabilities of GPS to non-GPS enabled nodes in a hop-by-hop fashion. Positioning is based on a hybrid method combining distance vector like propagation and GPS triangulation to estimate location in presence of signal strength measurement errors.

Ad hoc positioning is based on the following observation: if a node receives anchor information through multiple other nodes ( $N_1$  and  $N_2$  in Fig. 2.8) with known distance estimates ( $a$  and  $b$ , respectively) it calculates its own distance to that anchor. Because it is distance  $a$  away from  $N_1$  and distance  $b$  away from node  $N_2$  there can only be two possible locations for this node. See Fig. 2.8 for more detail. In this respect, APS assumes at least three nodes (called landmarks) that are GPS enhanced, or know their position by some other means.



**FIGURE 2.8** Ad hoc positioning visualized.

More accurate calculations can be achieved with a higher number of neighbors passing information about this anchor. With a third neighbor ( $N_3$ ) connected to either one of the previous neighbors, two circles can be drawn again with one other neighbor; this results in again a maximum of two intersections. With two pairs of two intersections, the same intersection should be in both pairs, which is then the correct intersection and that gives the node the distance to the anchor. If the third neighbor is connected to both previous neighbors the node should be able to determine whether or not it is on the same side as this new neighbor of the line  $N_1 \leftrightarrow N_2$ .

For the second phase, that is, determining location, Ad hoc positioning uses the lateration approach.

### 2.5.5.2 Using Directional Antennas for Location Estimation

Whereas omnidirectional antennas radiate isotropically in space, directional antennas focus (direct) the transmitting energy in the desired direction. Directionality is commonly achieved through a phased array. A prerequisite is that the elements of the phased array be an appreciable fraction of a wavelength apart. This is not feasible in electrically small form factor microdevices. However, limited directionality can be cheaply integrated using standard patch arrangements with high dielectric constant antennas.

The following paragraphs provide three different techniques to the problem of location estimation with directional antennas using one or more nodes with a known location.

**Directional Antenna Model** One of the simplest semi-directional antennas is the patch antenna. The ideal patch radiation model is a hemispherical radiator, which allows for semi-directional radiation. The typical gain of a patch antenna is on the order of 3.5–6 dBi, depending on the dielectric substrate used in the design. A representative angular

38 Chapter Two

variation of the gain for a typical microstrip antenna is in the range of  $\cos^2(\frac{\beta l}{2} \sin(\theta)) < G(\theta) < \cos(\frac{\beta l}{2} \sin(\theta))$ , where  $\beta$  is the free-space constant and  $l$  is the longest length of patch, assuming the lowest order mode of operation (Clarricoats et al., 1989). The gain is defined as the ratio of the intensity, in a given direction, to the radiation intensity that would be obtained if the power accepted by the antenna were radiated isotropically.

In the  $E$ -plane cut, the antenna's radiated e-field from a standard patch radiator is ideally  $E = \cos(\frac{\beta l}{2} \sin(\theta))$ . This pattern dependence is in relation to a coordinate system with the  $z$ -axis perpendicular to the microstrip patch radiator. This is the ideal solution for a patch antenna with an infinite ground plane and is only slightly altered using a finite size ground plane. The ground plane is used to shield the radiating field from the rest of the circuitry and the other radiators. Unshielded radiators, such as those that are standard with the Motes (Scott, 2004), are susceptible to parasitic radiating currents, which result in asymmetric patterns.

The received power at an antenna is given by  $P_r = \frac{P_t G_t(\Theta_t) G_r(\Theta_r)}{r^2} (\frac{\lambda}{4\pi})^2 \Sigma$ , where  $\Theta_t$  and  $\Theta_r$  are the transmitting and the receiving angles, respectively, and  $r$  is the distance between the transmitter and the receiver.  $\lambda$  is the RF wavelength of the carrier frequency.

Since  $(\lambda/4\pi)^2$  is a constant, we will exclude it from future expressions.

**Aligned antennas** In a number of practical applications it is reasonable to expect that meshes will be manually deployed. For instance, a WM<sup>2</sup>Net composed of mesh devices, which are set up to monitor a bridge's health, have to be placed by construction workers on the bridge. In such scenarios even though it may not be possible to know the precise location of the mesh node, it is possible to place these nodes in a predetermined orientation.

If the antennas of target nodes are aligned, then we can use the power received at multiple receiving antennas of the target from a single transmitting antenna on an anchor for position estimation. Without loss of generality, consider that an anchor node is placed to the southeast of the target node as shown in Fig. 2.9.

The size of the mesh node would usually be much smaller than the transmission distance. So  $d/r = \Theta_c$ . Then the received power at the two receiving antennas of the target node is given by Eqs. (2.7) and (2.8) in two variables  $\Theta_1$  and  $r$ . Since these are nonlinear equations it is difficult to get a closed form solution for  $\Theta_1$  and  $r$  in terms of

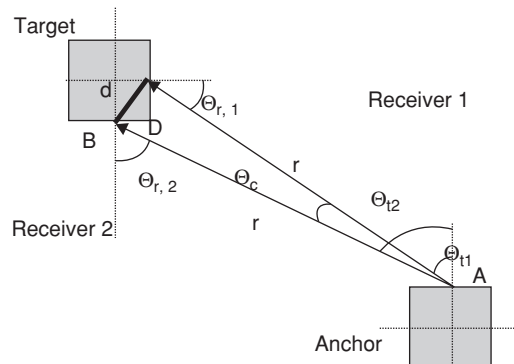
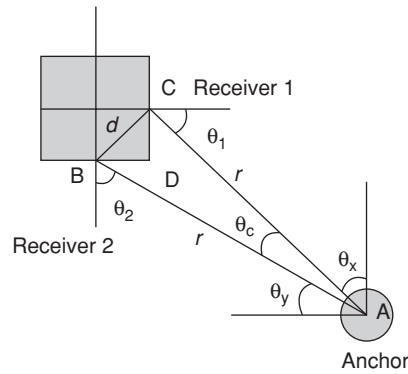


FIGURE 2.9 Location determination with aligned nodes.



**FIGURE 2.10** Location determination with an omnidirectional transmitter and directional receiver.

the input variables  $P_{r,1}$  and  $P_{r,2}$ . However, these equations can be numerically solved by standard methods to obtain  $\Theta_1$  and  $r$ :

$$P_{r,1} = \frac{P_t}{r^2} G_t(\Theta_{t,1}) G_r(\Theta_{r,1}) = \frac{P_t}{r^2} G_t\left(\frac{\pi}{2} - \Theta_1\right) G_r(\Theta_1) \quad (2.7)$$

$$\begin{aligned} P_{r,2} &= \frac{P_t}{r^2} G_t(\Theta_{t,2}) G_r(\Theta_{r,2}) = \frac{P_t}{r^2} G_t(\Theta_2) G_r(\Theta_2) \\ &= \frac{P_t}{r^2} G_t\left(\frac{\pi}{2} + \frac{d}{r} - \Theta_1\right) G_r\left(\frac{\pi}{2} + \frac{d}{r} - \Theta_1\right) \end{aligned} \quad (2.8)$$

where  $\Theta_1 = \Theta_{r,1}$  and  $\Theta_2 = \Theta_{r,2}$ .

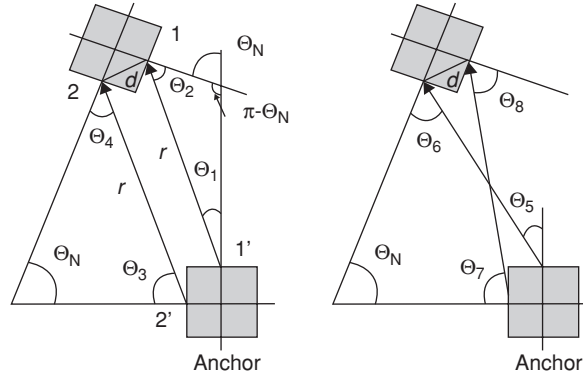
Alternatively, if the orientations of these meshes are not perfect,  $\Theta_1$  in Eqs. (2.7) and (2.8) can be replaced by  $\Theta'_1 = \Theta_1 - \Phi$  unaligned, where  $\Phi$  unaligned can be obtained from a digital compass (*DMC-SX Digital Magnetic Compass—Operator Manual*, Leica Vectronix AG, Switzerland) or some other simple algorithms (Fang et al., 2005). A possible approach is to mount an omnidirectional antenna with the four directional antennas on the same node and estimating  $\Phi$  unaligned from the difference of the received power strength between the directional antennas and the omnidirectional antenna.

A baseline experiment for this is with the anchor node having omnidirectional dipole antennas. In this case, the gain of the transmitter,  $G_t(\Theta)$ , is constant over all  $\Theta$  and denoted  $G_{\text{omni}}$ . Figure 2.10 shows this configuration. Now, since we know the distance as well as the relative direction of the target with respect to the anchor, we can estimate its position.

The estimates from multiple anchors can be averaged to obtain a better estimate of the position. Alternatively, the information about  $\Theta_1$  could be discarded and the range measurements ( $r$ ) can be used to triangulate the position of the node in a least squares manner. Both these strategies have been evaluated using computer simulations. The results showed that the averaging strategy yields better results.

**Generalization to Unaligned Antennas** In cases where it is not possible to ensure a global orientation of all nodes of a network, additional measurements can be used to estimate position. Received power at two different antennas of the target node from two

## 40 Chapter Two



**FIGURE 2.11** Location determination for unaligned antennas.

transmitting antennas of the anchor node is measured. Such an arrangement is shown in Fig. 2.11.

Geometric relations between the various transmission and receiving angles can be derived from Fig. 2.11.

$$\begin{aligned} \Theta_2 + \Theta_6 &= \Theta_4 + \Theta_8 = \Theta_3 + \Theta_5 = \Theta_1 + \Theta_7 = \frac{\pi}{2} + \frac{d}{r} \\ \Theta_1 + \Theta_2 + \Theta_3 + \Theta_4 &= \pi \end{aligned} \quad (2.9)$$

$$\begin{aligned} P_{r,11'} &= \frac{P_t^* G_t (\pi - \Theta_2 - \Theta_3 - \Theta_4)^* G_r (\Theta_2)}{r^2} \\ P_{r,21'} &= \frac{P_t^* G_t \left(\frac{\pi}{2} + \frac{d}{r} - \Theta_3\right)^* G_r \left(\frac{\pi}{2} + \frac{d}{r} - \Theta_2\right)}{r^2} \end{aligned} \quad (2.10)$$

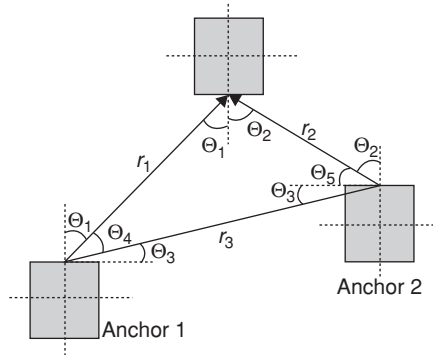
Let  $P_{r,ij}$  denote the power received by antenna  $i$  on the target node when antenna  $j$  is transmitting on the anchor node. We can use these equations to simplify the received power equations as follows.

$$P_{r,12'} = \frac{P_t^* G_t \left(\frac{d}{r} + \Theta_2 + \Theta_3 + \Theta_4 - \frac{\pi}{2}\right)^* G_r \left(\frac{\pi}{2} + \frac{d}{r} - \Theta_4\right)}{r^2} \quad (2.11)$$

$$P_{r,22'} = \frac{P_t^* G_t (\Theta_3)^* G_r (\Theta_4)}{r^2} \quad (2.12)$$

Equations (2.9) to (2.12) in the four variables  $\Theta_2$ ,  $\Theta_3$ ,  $\Theta_4$ , and  $r$  can again be numerically solved to estimate the location of the target node.

This scheme requires that two target antennas be able to simultaneously receive transmissions from two anchor antennas. This would require a transmitter beam width of  $180^\circ$ . This is nonoptimal for four antennas covering a  $360^\circ$  plane but is a tradeoff for increased degrees of freedom in the orientation of nodes. Besides, the increased beam-width will lend greater fault tolerance to the system by providing greater redundancy in the areas reached by multiple transmitting antennas. It will also make the antenna design easier since high directionality, that is, narrow beam width is not needed.



**FIGURE 2.12** Location determination using measurements from two anchors.

**Aligned Antennas with Two Anchors** The two location determination methods described above rely on the difference in power received at two antennas of a node from the antennas on the same anchor node. The error in the power received can become correlated due to the proximity of the two antennas, even if they are pointed in separate directions. In a real life scenario the correlation can significantly reduce the accuracy of the location estimate, especially for a very small mesh node. To investigate the performance of the location determination with increasingly uncorrelated channels, two transmitted signals were sent from two nodes substantially positioned far enough from each other. This scheme is also useful in situations where more than one directional antenna would not fit on a single node. The arrangement is shown in Fig. 2.12.

Since the location of the two anchors is known, the parameters  $r_3$  and  $\Theta_3$  can be determined. Using geometric properties of the system, we get the following relations between the various angles

$$\Theta_5 = \frac{\pi}{2} - \Theta_2 \quad \Theta_1 + \Theta_3 + \Theta_4 = \frac{\pi}{2}$$

The equations for the received power are given by

$$P_{r,1} = \frac{P_t}{r_1^2} G_t(\Theta_1) G_r(\Theta_1) \quad (2.13)$$

$$P_{r,2} = \frac{P_t}{r_2^2} G_t(\Theta_2) G_r(\Theta_2) \quad (2.14)$$

Using the law of sines along with relations between the angles derived earlier we get two more equations

$$\frac{r_2}{\cos(\Theta_1 + \Theta_3)} = \frac{r_3}{\sin(\Theta_1 + \Theta_2)} \quad (2.15)$$

$$\frac{r_1}{\cos(\Theta_2 - \Theta_3)} = \frac{r_3}{\sin(\Theta_1 + \Theta_2)} \quad (2.16)$$

This gives us four equations with four unknowns:  $r_1, r_2, \Theta_1$ , and  $\Theta_2$ . Thus, the distance and the angle with respect to each of the two anchors are determined. The mesh node's

## 42 Chapter Two

location can be estimated using either distance or angle pair or the final location that is estimated averaging out all estimates.

### 2.5.5.3 Methodologies for Geographic Forwarding

There exist three common strategies for geo-forwarding: **greedy forwarding**, **directed flooding**, and **hierarchical routing**. For the first two strategies, a node with a packet to relay forwards it to one (greedy forwarding) or more (directed flooding) one-hop neighbors that are located closer to the destination than the forwarding node itself. The position of the neighboring nodes is typically learned through a periodic beaconing scheme. That is, all nodes periodically broadcast their one-hop beacons that contain their latest position information. Using this information, the recipient nodes update their own positions, and so on. Each node thus maintains a table with the positions of all its direct neighboring nodes. The third forwarding strategy aims at structuring mesh nodes into a hierarchy. Hierarchical mechanisms may use different types of routing protocols at different levels of the hierarchy (e.g., a nonposition-based routing protocol at one level and a position-based protocol at a different level).

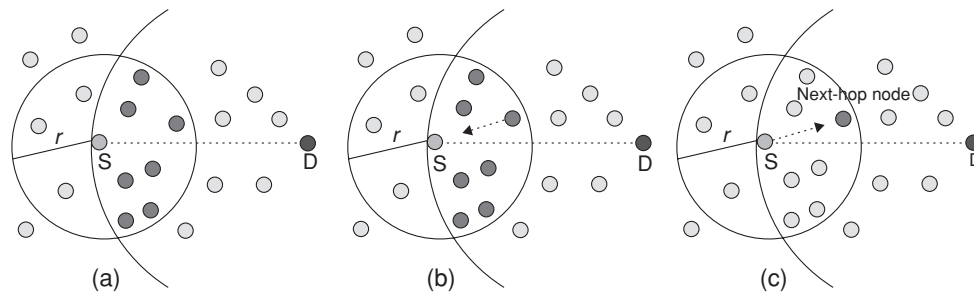
The position information calculated from the location service is included in the header of the packet for forwarding decisions. Intermediate nodes may not need to consult their location table to obtain a more accurate position of the destination, but simply route the packet using the location information carried in its header. However, if an intermediate node maintains a more accurate position for the destination, it may well choose to update the position information in the header of the packet prior to forwarding it.

### 2.5.5.4 Contention-Based Geographic Forwarding: A Communication Paradigm for Efficient Data Delivery in WM<sup>2</sup>Snets<sup>2</sup>

#### Contention-Based Geographic Forwarding (CGF)

**Traditional Geographic Forwarding** Traditional geographic forwarding encapsulates two modes of operation: *greedy forwarding* and *void handling*. In the greedy forwarding mode, the forwarding decision is based on the location of the sender, the location of the destination, and the locations of the sender's neighboring nodes. The location of the destination is attached in the header of the packet. An intermediate node that receives the packet checks the location of the destination in the header of the packet prior to forwarding it. In case the node maintains more accurate location information for the destination in its database, it is then free to update the location information in the header of the packet. The location of the neighboring nodes is typically learned through a periodic beaconing scheme. That is, all nodes periodically update their own locations and broadcast their one-hop beacons that contain their latest location information. The beaconing scheme is a proactive component and independent of actual data traffic and network dynamics. Nodes maintain a table with the location of their one-hop neighbors as well as of distant nodes. If a sender cannot locate a next-hop node with a positive progress towards the destination, called a *communication void*, it switches to the void handling mode in order to route the packet to around the void.

<sup>2</sup>Excerpt from the invited article "Contention-Based Geographic Forwarding: A communication paradigm for efficient data delivery in wireless mesh networks," Dazhi Chen and Pramod K. Varshney, Department of Electrical Engineering and Computer Science, Syracuse University Syracuse, New York, NY 13244, USA.



**FIGURE 2.13** A high-level description of CGF. (a) The sender broadcasts a request to its one-hop neighboring nodes. (b) The winner from potential next-hop candidate nodes in the forwarding area confirms with the sender. (c) The sender forwards actual data packet to the established next-hop node.

The proactive nature of the beaconing mechanism induces several limitations, which are observed from three perspectives. First, the periodic transmission of beacons consumes a considerable amount of energy and wireless bandwidth resources unnecessarily. When no data are exchanged for long periods of time, beacons are sent unnecessarily. Second, in a highly dynamic network, a neighboring node selected as a next-hop node may no longer be available before beacons update such information. This leads to significant packet losses and hence to a very high communication overhead due to MAC layer packet retransmissions resulting in increased delivery ratios. In order to obtain more accurate position information, it is likely to increase the beaconing rate. However, this would result to significant increase of communication overhead. On the other hand, in a relatively static network, where the network infrastructure remains unchanged for long periods of time, beacons are still sent out at periodic intervals.

**Contention-Based Geographic Forwarding** To address the aforementioned performance concerns, a new geographic forwarding paradigm, called CGF<sup>3</sup> (Chen et al., 2005), is proposed aiming to eliminate the beaconing scheme. In contrast to traditional geographic forwarding, a distributed contention-based mechanism is used in CGF to select a next-hop node among a group of neighboring nodes.

Figure 2.13 illustrates the basic procedure of the CGF paradigm. To forward a data packet, a sender broadcasts a control packet requesting its one-hop neighboring nodes to relay its data. The neighboring nodes that are eligible to forward the data packet, for example, they are within a forwarding area which is expected to make a positive geographic progress towards the destination, are called next-hop candidate nodes. The sender then unicasts the actual data packet to the next-hop node that won the contention. If a communication void occurs, that is, if no next-hop node appears in the forwarding area, a void handling scheme is employed to route the packet around the void. This process is repeated until the packet is successfully delivered to its destination.

Based on the high-level description, CGF mainly consists of the following components:

<sup>3</sup> It is also called random forwarding, implicit forwarding, volunteer forwarding, and contention-based forwarding (Zorzi and Rao 2003; Blum et al. 2003; Chen et al., 2005a; Füßler et al., 2003; Heissenbüttel et al., 2004; Ferrara et al. 2005, Xu et al., 2005).