Chapter 2
Routing for Wireless Multi-Hop Networks: Unifying Features

Abstract Wireless multi-hop networks share some routing features based on the fact that they all follow the multi-hopping paradigm. In this chapter, we follow a component-based approach for breaking down a routing protocol into some core and auxiliary components. We discuss the core components that are fundamental for any wireless multi-hop routing protocol along with some auxiliary components that can be adopted to achieve a specific design goal. Dependency and relationships among the components are elaborated as well. Finally, we propose a generic routing model that can be inherited for the design of any wireless multi-hop routing protocol.

Keywords Wireless multi-hop networks · Route discovery · Route selection · Route representation · Data forwarding · Route maintenance · Route energy efficiency

2.1 Introduction

Routing is the main function of the network layer, the 3rd layer of the protocol stack, and its performance is highly affected by the lower layers: the physical and data link layers. In order for a routing protocol to be efficient and reliable, the protocol designer should consider the effects of the lower layers and provide mechanisms for handling these effects. For example, due to some features of the physical layer, the communication range of the devices/communicating nodes may be asymmetric. This means that if node A can send a message directly to node B, it is possible that node B cannot reply back directly to node A. These communication issues should be taken into account when designing a routing protocol. On the other hand, this kind of cross layer effect can be utilized to improve the

S. Abdel Hamid et al., *Routing for Wireless Multi-Hop Networks*,
SpringerBriefs in Computer Science, DOI: 10.1007/978-1-4614-6357-3_2,

performance of the routing protocol. For example, routing protocols that are designed to support QoS and low latency requirements must consider link qualities in choosing the optimal path among the set of available paths. These link quality measures can be obtained from the lower layers by passing parameters to the network layer. A designer of a routing protocol then should consider the functionalities of the lower layers, handle their affecting features, and utilize their measures, parameters, and, in some cases, their layer-specific packets.

For the wireless multi-hop networks, MANETs, WSNs, WMNs, and VANETS, there is no single wireless multi-hop routing protocol which can fit all needs. This is because each network paradigm has its own design challenges. Yet, as they all are classified under the category of wireless multi-hop networks, they have some unifying features. There are some routing functionalities and components that are essential, and are common parts of any wireless multi-hop routing protocol.

In Ref. [1], Lee et al. proposed a taxonomy that can be followed in designing a wireless routing protocol. They propose breaking down the wireless routing protocol and functions into multiple smaller components. Some of these components are core ones that should be a part of any wireless routing protocol and others are auxiliary that can be included only when needed by the application requirements. Following this component approach, in the following section, we will provide a detailed discussion of the routing components showing the core and auxiliary ones, and when these auxiliary ones may be needed.

2.2 Routing Components: An Exhaustive View

By breaking down the wireless routing protocol into smaller components, we can analyze the components that should be included in any wireless multi-hop routing protocol and show the interacting behavior between them. The behavior of these basic components can be tailored to different application profiles and needs, while keeping and maintaining the core functional behavior and goals [1]. To satisfy network and application specific needs, extra components can be added to the routing protocol to control its behavior and maintain its performance as needed and specified by the application and network paradigm. Having the core components, a routing protocol can be easily extended to accommodate and support extra requirements, services and features by adding auxiliary components. In the two following sub-sections, we will discuss the core components that should be a part of the skeleton of any routing protocol and we will shed light on some auxiliary components that may be used only based on the network and application needs.

2.2.1 Core Components

These components are considered to be basic building blocks for any wireless routing protocol to provide its main function, getting a message from a source to a destination. These components are route discovery, route selection, and route representation and data forwarding.

2.2.1.1 Route Discovery

Route discovery is the first stage of the function of any wireless routing protocol. Route discovery is the process of finding a route/set of potential routes between a source and an intended destination. The process of finding a route can be classified into three categories: *proactive*, *reactive* or *hybrid*.

Proactive route discovery, also known as *table-driven* route discovery, depends on the use of up-to-date routing information about the whole network to find a path from any source to any destination in the network. This routing information is exchanged among nodes either periodically or upon the occurrence of any change in the network topology. This information is kept at each node in a routing table. This type of route discovery pre-determines routes between any two nodes irrespective of the need for such routes. When a node has a packet to be sent, it does not need to wait for a route to be discovered. It consults its routing table, gets the up-to-date recorded route, then sends the packet without incurring a delay for the route to be discovered—the route is discovered a priori.

There are two sub-categories under the proactive routing category: Distance Vector (DV) and Link State (LS). They differ in how the network topology information is spread. These techniques are borrowed from wired networks but they can be modified to handle the characteristics of MANETs.

(a) **Distance Vector Proactive Routing**

In DV route discovery, each node maintains a routing table where it stores information about all possible destinations, the next node to reach that destination, and the best known distance to reach the destination.[1]

These tables are updated by exchanging information with the neighbors. Each node periodically sends a vector to its direct neighbors carrying the information recorded in the routing table to maintain topology. The distance vector contains the destinations list and the cost—the distance—to reach each destination.

The basic distance vector routing technique works in theory but has a serious drawback in practice. It suffers from a severe problem known as "count-to-infinity" [2]. This happens as a result of the occurrence of routing loops; when X tells Y that it has a path somewhere, Y has no way of knowing

[1] Distance can be defined as the number of hops.

whether it itself is on the path. This drawback is common in the DV routing technique, and all DV-based routing protocol designers should consider this issue and find a mechanism to avoid it.

(b) **Link State Proactive Routing**

Distance vector routing was used in ARPANET until 1979, when it was replaced by link state routing. The objective of LS routing is to provide an alternative to DV that avoids routing loops and the subsequent "count-to-infinity" problem. LS routing overcomes this by maintaining global network topology information at each node.

In LS routing, each node periodically sends information about the cost to reach each of its direct neighbors and it includes this information in what is known as the link state packet. This link state packet is sent to all the other nodes in the network by flooding. Each node does the same link state flooding procedure and, eventually, each node will have link state packets from all other nodes, so each node will have information about the complete topology and costs of all the links in the network. Then Dijkstra's algorithm [3] can be run locally to construct the shortest path to all possible destinations. The results of this algorithm can be stored in the routing tables for later use [2].

Although LS routing avoids some problems with DV routing, it has a problem with its storage requirements.

As an advantage, proactive route discovery incurs almost no delay as routes are calculated in advance and are available in the routing table. However, it has a disadvantage that may hamper its use in large networks. It incurs an overhead related to the periodic routing updates which may cause congestion for the network when it has a large number of nodes. Therefore, in most cases, the proactive route discovery has problems with network scalability.

Reactive route discovery is also known as *on-demand* route discovery. As the name implies, the route is discovered on demand. When a source has a packet to be sent, it initiates a route discovery process to set up a path to the intended destination. Many approaches can be followed for path setup where the most common one is having the source node broadcast a route request packet carrying the destination address and asking for a route to that destination. When the route request reaches the destination or an intermediate node that knows a route to that destination, a route reply packet is sent back to the source carrying details about the discovered route.

Some protocols perform route discovery on the fly, hop-by-hop. When a node receives a packet to be forwarded to another node, it decides to which neighbor it should forward this packet. This type of routing is known as self-routing and it falls under the category of reactive routing as the route is established on demand. An example of this type of routing is geographical routing where a node picks the next hop based on the locations of its neighbors and their distances to the destination. The self-routing based protocols usually require a form of neighbor discovery to know about the potential forwarding nodes that the current node will choose from to be the next hop.

Reactive route discovery avoids the drawback of the proactive one by avoiding exchanging periodic routing updates, which reduces the traffic overhead. However, as the path is discovered only on-demand, this type of route discovery incurs a delay overhead and longer latency for route establishment.

The category of ***hybrid*** route discovery is obtained by combining both the proactive and reactive techniques to make use of the advantages of both and mitigate their disadvantages. It tries to reduce the control overhead associated with proactive route discovery and the delay incurred in the reactive one.

2.2.1.2 Route Selection

As an output of the route discovery stage, there will be a set of potential routes between a source and destination. It is the role of the route selection component to pick the optimal path from this set that satisfies the needed performance criteria. Most of the routing protocols are based on choosing only one path for delivering packets from a specific source to a specific destination; however, there are some protocols that rely on choosing multiple paths (multipath routing) [4] to provide redundancy and fault tolerance for the routing process.

For the proactive protocols, route selection is done implicitly with the route discovery stage. When the network topology information is shared and received by the nodes, they update the information in their routing tables accordingly; hence, routes available in the routing tables are the selected, best ones at that time.

Route selection in the reactive protocols is a stand-alone process. It can be handled by the source, the destination, or the intermediate nodes. In *destination-based* route selection, when the destination receives multiple route requests forwarded by multiple intermediate nodes, it can select the path to receive data through and sends the route reply along this path. The destination can pick the first path through which it received the first route request, the fastest one, or it can wait for a specific period of time. Thereafter, if it has received many route requests, it can pick the optimal path according to some selection metrics, discussed later in this section.

In *source-based* route selection, the source node may receive multiple route replies from the destination,[2] or from all intermediate nodes that know about a route to the intended destination. It is the responsibility of the source to pick a route from the set of routes extracted from the multiple route replies.

For *intermediate-based* route selection, the intermediate nodes decide on which route a packet should follow to reach a destination. They can either choose a route from a set of possible routes they keep for that destination,[3] or select a next hop on the fly. This type of route selection is involved in self-routing protocols. Since the

[2] The destination may reply to all route requests it receives.

[3] These routes may be discovered by them in a previous interaction with the destination or overheard from neighbors interacting with that destination.

route is selected on a hop-by-hop basis, the intermediate nodes are involved in the route selection process when the packet is forwarded to any of them.

Whether route selection is done by source, destination, or intermediate nodes, the deciding node should depend on one or several metrics for the selection decision. Determination of which route metric to use is dependent on the application requirements and needs. The simplest route metric and the most popular one is the hop count. The path with the least hop count will be chosen to reduce the number of intermediate nodes involved in the routing process and so reduce the control overhead and contention level among nodes. Examples of other routing metrics include energy consumption level, residual energy of the next hop, QoS metrics (such as end-to-end delay/jitter, interference level, packet loss rate, link residual capacity, and load balancing), link security level, and memory consumption. Some of these routing metrics require parameters related to the lower layers like the QoS-based link quality ones. These parameters can either be passed from the lower layers to the routing layer, or, in some protocols, this interaction with the lower layers is done in the form of cross-layer protocol design.

In short, how the route is selected is based on the application/network paradigm for which the protocol will be used. It is how the route will be selected that controls the performance of the routing protocol and whether it will satisfy the needs of the application or not.

2.2.1.3 Route Representation and Data Forwarding

After selecting a route, it should be stored to be followed for data transfer. We consider both route representation and data forwarding as a single component as they are highly integrated together and, in many protocols, they are done simultaneously. Route representation and data forwarding can follow one of two techniques: *exact route* and *route guidance* [1].

(a) **Exact Route**

In this technique, the sequence of intermediate nodes that a path should follow to reach a destination is represented explicitly. There are two approaches for using the exact route representation and forwarding. These approaches are routing table and source routing.

- *Routing Table*

  In this approach, each node keeps a routing table where it stores the next hop to reach potential destinations with one entry per destination. In the proactive protocols, this routing table contains information and next hops to all other potential destinations in the network. In the reactive protocols that make use of the routing table approach, they keep information about the destinations that they interacted with previously or those nodes that they overheard paths to them. Also, in these routing tables, they may keep information about nodes from which they received route requests or route replies for further relaying. When a packet is to be forwarded, the node looks up the routing table and gets the next hop to which it should forward the packet to reach the intended destination.

- *Source Routing*
  The idea of the source routing approach is to include the whole path that the packet should follow in the packet header, and when a node gets this packet, it can extract the next hop from the path included in the packet. This path is included by the source node before sending the packet. This approach encounters some problems especially with large networks; as the complete path is included in the packet header, this can be considered traffic overhead and a source for bandwidth wasting.

(b) **Route Guidance**
In route guidance-based protocols, the sequence of intermediate nodes is not explicitly described. The full path is not determined prior to sending the packet by the source, rather the path is formed on the fly (i.e., self-routing). As the route is not fully determined a priori, nodes cannot store information about the path itself but they may store information about how the next hop will be chosen or information that will be used for picking the next hop. This is what is called route guidance. The geographical routing protocols are examples of protocols that follow the route guidance technique. In these protocols, instead of keeping information about the path itself, nodes store the positions of their neighbors and pick the next hop on the fly based on the destination and their direct neighbors' positions [5].

The three aforementioned components are considered core ones that should be included in any wireless routing protocol. As mentioned above, their behavior can be tailored to meet the requirements of the network paradigm that the protocol is designed for and this will be discussed in Chap. 3. In the following section, we will explore some of the auxiliary routing components that can be added to the core components to achieve a specific design goal.

2.2.2 Auxiliary Components

These components are not essential for all routing protocols but they can be added to improve the performance of a protocol or to make it meet the requirements and needs of a specific application or network paradigm. Examples of these components are route maintenance, route energy efficiency, and route security. Some of these components are discussed in the following.

2.2.2.1 Route Maintenance

The goal of the route maintenance component is to keep a route valid while in use and to handle possible failures. Route maintenance is needed by networks where links are prone to failure due to node mobility, for example. It is considered a crucial component in MANETs where nodes are highly mobile and the network topology encounters frequent dynamic changes. Route maintenance includes *route refreshing*, *route failure handling*, and *route invalidation* [1].

(a) **Route Refreshing**
Route refreshing aims at keeping the current routes valid by updating or using them only for the sake of refreshing. Route refreshing can be handled by one of three approaches depending on the category of the protocol: proactive, reactive, or hybrid. In the proactive protocols, route refreshing is done implicitly by having the nodes periodically or upon the occurrence of topology changes exchange network topology information and update the routing tables according to the current changes in the network. Therefore, in the proactive protocols, routes in the routing table are always the most up-to-date. In the reactive approach, routes are only touched on demand, so to keep routes usable and ensure their validity, nodes can refresh routes either by use of control packets (e.g., hello messages) or by using a data packet before the expiration of the route. Hybrid protocols and hybrid route refreshing combine both the proactive and reactive approaches.

(b) **Route Failure Handling**
In reactive routing, when an intermediate node finds that the next hop is unreachable, it tries one of two options: (1) to find an alternate path locally either by looking up its routing cache for an alternative or by initiating a route discovery process to replace the failing link with a valid one, or (2) to send a route error message to the source node with information about the failing link. The source node can also look up its route cache for a different route. If there is no alternative, it reinitiates the route discovery process while marking the failing part in order not to include it again.
In proactive routing, route failure is handled by route refreshing. As the routing tables have up-to-date routing information, route failure is handled by automatic updates.
In the hybrid protocols, it is a combination between the proactive and reactive route failure handling approaches.

(c) **Route Invalidation**
Route invalidation is the process of finding out stale routes and removing them from the routing tables and caches. The stale routes are distinguished and recognized by employing a lifetime period for each route, and if this route has not been refreshed during that period, it will be marked as expired and will be removed.

2.2.2.2 Route Energy Efficiency

As some of the wireless multi-hop networks are comprised of devices with limited resources, e.g., sensor nodes in WSNs, such networks have energy efficiency as one of the major design considerations that should be taken care of in any protocol designed for such networks including the routing ones. Routing protocols designed for such networks should include mechanisms to conserve node energy to prolong the lifetime of the nodes and of the network as a whole. Examples of such techniques are data aggregation, use of meta-data, load balancing, restricted flooding, use of energy-aware metrics, use of a resource manager, and putting nodes into sleep mode.

(a) **Data Aggregation**
Data aggregation is one of the techniques that is highly utilized in the energy-efficient routing protocols because, when deployed, it has a great impact on the nodes' residual energy and lifetime. The idea is that instead of sending redundant packets or packets that have a kind of correlation, these packets can be combined and aggregated together into only one packet. Reducing the number of transmitted packets leads to great conservation in node energy.

(b) **Use of Meta-Data**
A number of protocols depend on sending meta-data that describes the actual data packets instead of sending the actual packets themselves. This technique is mainly used for advertising the actual data. Instead of sending long data packets to nodes that may not be interested in them, small meta-data is sent to advertise the acquired data packets and if a node shows its interest in such data, the complete data packet is sent to it afterwards.

(c) **Load Balancing**
Many protocols focus on balancing the traffic load among the nodes in order not to overload some nodes compared to others which may lead to depletion of these nodes' batteries and cause their failures. For example, in cluster-based routing protocols, if cluster formation is static and not changed throughout the network life, the nodes that act as cluster-heads will burn their energy quickly, and after they die, all their members will be "headless" and therefore useless. This is because the role of being a cluster-head is energy consuming as the cluster-head has to be awake all the time, receive data from all of its cluster members, incur processing overhead for aggregating the data, and is responsible for the long-range transmissions to the data collector. To provide energy efficiency and balance energy consumption among the nodes, some routing protocols utilize dynamic clustering to rotate the role of being a cluster-head among the nodes.

(d) **Restricted Flooding**
When a packet needs to be broadcast (e.g., route request packets or data interests), some protocols make use of restricted flooding instead of flooding the packet to the whole network. For example, the packet can be sent to a group of nodes with higher probability to forward the packet or with wider coverage and view for the network. Another example is forwarding the packet to an area of interest instead of to the whole network, for example, sending data interests geographically to the area of interest then flooding the interest only within this area.

(e) **Use of Energy-Aware Metrics**
When this technique is utilized, it can be considered a part of the route selection component. To conserve energy, the optimal route can be selected based on the energy of the nodes constituting that route. A node's current energy consumption level or current residual energy can be used as the route selection metric.

(f) **Use of a Resource Manager**
Some protocols add to the routing component a resource manager that monitors the energy level of the nodes and adjusts their operations based on some thresholds.

(g) **Putting Nodes into Sleep Mode**
As a common technique in most of the WSN protocols (either MAC, routing, or other layer protocols), putting nodes into sleep mode saves a significant amount of energy. In the sleep mode, only the processor works with only a small portion of its capabilities; neither sensing nor transmissions are done. Once the node gets tasked or awakened, it works with all its capabilities.

2.3 Generic Routing Model

In this section, we will present a generic routing model that can be used to form the foundation of a wireless multi-hop routing protocol. We will present the functionalities as blocks and methods that can be selectively utilized and combined together to form a wireless routing protocol suitable for any wireless multi-hop network. This generic model can be further extended and enhanced with auxiliary functionalities to meet specific requirements per network paradigm.

Each component will be presented with its own various functionalities that will be available to the protocol designer to choose from. The output and the input of each component will be shown to clarify the interactions between the various components. The proposed generic model is shown in Fig. 2.1.

The route discovery component has five options/functions for the designer to choose from: (1) proactive with distance vector, (2) proactive with link state, (3) reactive with deterministic routing, (4) reactive with self-routing (which requires that each node discovers its neighbors; therefore, it calls the neighbor discovery function which feeds it with the neighbors list), and (5) hybrid discovery.

The route selection component has three functions for the protocol designer to choose from: (1) source-based selection, (2) destination-based selection, and (3) intermediate-based selection. The choice of which function to be used depends on the route discovery function that has been chosen (e.g., the reactive self-routing discovery requires the use of intermediate-based route selection).

Finally, the route representation and data forwarding component has three functions available for the designer's choice: (1) representation and forwarding

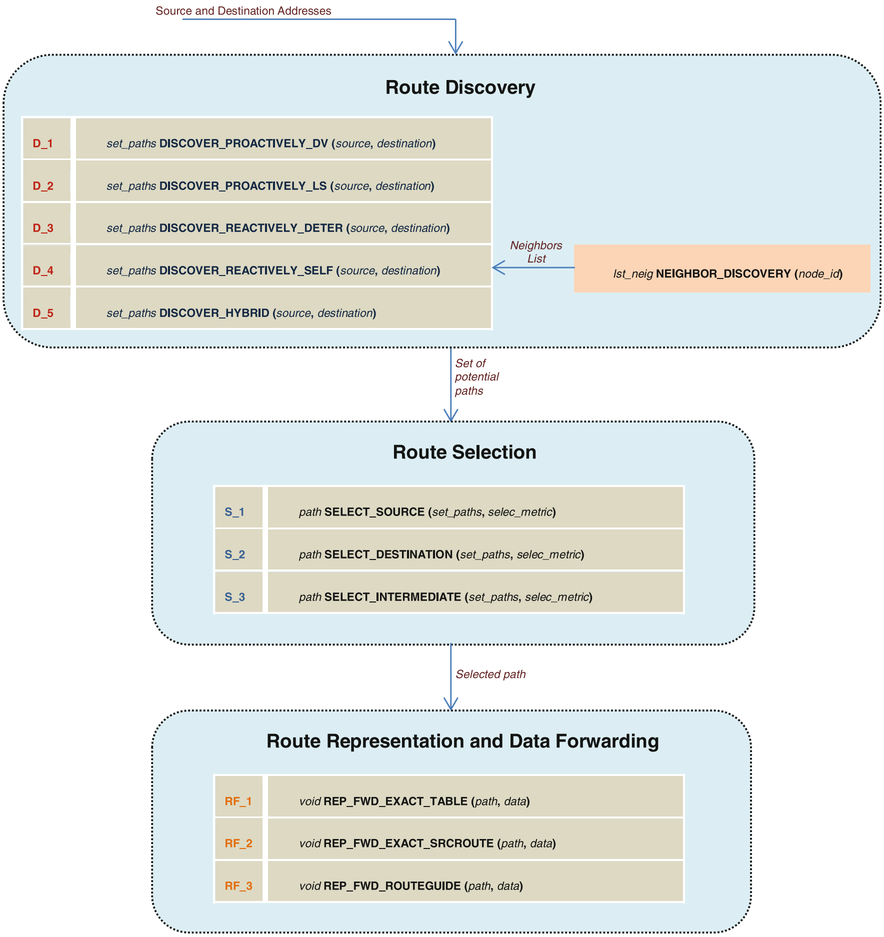

Fig. 2.1 Generic routing model

using exact route with routing tables, (2) representation and forwarding using exact route with source routing, and (3) representation and forwarding using route guidance. Again, the choice of the appropriate function strictly depends on the chosen discovery function (e.g., the reactive self-routing discovery requires the use of route guidance).

The following pseudo-code shows the interaction and dependency of the route selection function and the route representation and data forwarding function to be chosen and the already chosen discovery function. For simplicity, we refer to the functions by codes—these codes are shown in Fig. 2.1 next to their associated functions.

By breaking down the functionalities into blocks and methods, the protocol designer can choose whatever functionalities are preferred and suitable for the intended paradigm and application. In addition, the designer can replace the chosen functionality of each component without having to redesign the whole protocol. The protocol design is based on a set of blocks that can be edited separately.

In implementing or modifying any of the functionalities, the protocol designer should consider including the scalability[4] and self-configuration[5] features as they are both basic features for all wireless multi-hop routing protocols.

Pseudo-code for choosing the route selection function and the route representation and data forwarding function based on the chosen discovery function

```
if D_1 or D_2 is chosen then
    choose S_1 and RF_1
else if D_3 is chosen then
    choose S_1 or S_2 or S_3 and RF_1 or RF_2
else if D_4 is chosen then
    choose S_3 and RF_3
else if D_5 is chosen then
    // For the proactive part
    choose S_1 and RF_1
    and
    // For the reactive part
    if D_3 is chosen then
        choose S_1 or S_2 or S_3 and RF_1 or RF_2
    else if D_4 is chosen then
        choose S_3 and RF_3
    end if
end if
```

2.4 Summary

In this chapter, we discussed the unification of multi-hop networks in terms of their routing functions. We followed a component-based approach for breaking down a routing protocol into some core and auxiliary components. We presented the core components that are considered a part of any wireless multi-hop routing protocol and are considered the common and unifying features of all wireless multi-hop

4 Since the network is ad hoc, the number of nodes can always increase.

5 Note that there is no central control.

routing schemes. As well, we discussed some auxiliary components that can be added to the core ones to achieve a certain design goal. Finally, we introduced a generic routing model that can be inherited by, and considered the basis of, any wireless multi-hop routing protocol.

References

1. Lee MJ, Zheng J, Hu X, Juan H, Zhu C, Liu Y, Yoon JS, Saadawi TN (2006) A new taxonomy of routing algorithms for wireless mobile ad hoc networks: the component approach. IEEE Communications Magazine 44 (11):116–123. doi:10.1109/MCOM.2006.248174
2. Tanenbaum AS (2003) Computer Networks. 4 edn. Prentice Hall,
3. Dijkstra EW (1959) A note on two problems in connexion with graphs. Numerische mathematik 1 (1):269–271. doi:10.1007/BF01386390
4. Mueller S, Tsang R, Ghosal D (2004) Multipath routing in mobile ad hoc networks: Issues and challenges. Performance Tools and Applications to Networked Systems:209–234
5. Mauve M, Widmer A, Hartenstein H (2001) A survey on position-based routing in mobile ad hoc networks. IEEE Network 15 (6):30–39. doi:10.1109/65.967595

Springer

http://www.springer.com/978-1-4614-6356-6

Routing for Wireless Multi-Hop Networks
Abdel Hamid, S.; Hassanein, H.S.; Takahara, G.
2013, XV, 73 p. 14 illus., Softcover
ISBN: 978-1-4614-6356-6