

Routing and Security in Mobile Ad Hoc Networks



Manets offer a promising new wireless communications paradigm, but researchers must develop efficient routing algorithms and address security concerns before such networks can be extensively deployed.

Nikola Milanovic
Miroslaw Malek
Humboldt University

Anthony Davidson
New York University

Veljko Milutinovic
University of Belgrade

Wireless technologies such as General Packet Radio Service, Wi-Fi, HomeRF, and Bluetooth make it possible to access the Web from mobile phones, print documents from PDAs, and synchronize data among various office devices. However, such applications rely at some point on mobility support routers or base stations, and it is often necessary to establish communication when the wired infrastructure is inaccessible, overloaded, damaged, or destroyed.

Mobile ad hoc networks remove this dependence on a fixed network infrastructure by treating every available mobile node as an intermediate switch, thereby extending the range of mobile nodes well beyond that of their base transceivers. Other advantages of manets include easy installation and upgrade, low cost and maintenance, more flexibility, and the ability to employ new and efficient routing protocols for wireless communication.

We present four manet routing algorithms along with a hybrid approach, discuss their advantages and disadvantages, and describe security problems inherent in such networks.

MANETS

Suppose that we want to easily and efficiently connect two office floors using short-range wireless communication devices. Every employee has one of these mobile devices, and some fixed devices—computers, printers, and so on—have the same capability.

We could connect these devices to the existing wired infrastructure using access points, but this

option offers limited mobility, adds load on the wired network, and relies on existing protocols for wired communications. Another possibility is to build a network of dedicated and mutually connected base stations that enable cellular communication, but this is expensive with respect to time, installation, and maintenance.

The best solution is to create a mobile ad hoc network using surrounding electronic devices as intermediate switches when they are idle and if they are capable of performing this task. For example, the packet from one device can hop to the mobile phone of a person passing through the corridor in front of the office, then from the mobile phone to the shared laser printer in the next office, then to someone's digital wristwatch on the floor below, then from the wristwatch to the coffee machine, and, finally, from the coffee machine to its ultimate destination—say, another colleague's device or computer.

Manets are also useful for disaster management. A communications infrastructure is designed to survive common short-term problems, such as overloading, but not to sustain major physical damage. In most cases, the collapse of a single system will cause many dependent devices to fail. If a fire, earthquake, or other natural catastrophe disables a subset of base stations, every mobile phone within range of those stations automatically becomes unreachable. In such situations, rescue workers can use the nodes in manets to create a network “on the fly.”

Small-scale manets are also effective for emergency search and rescue, battlefield surveillance, and other communication applications in haz-

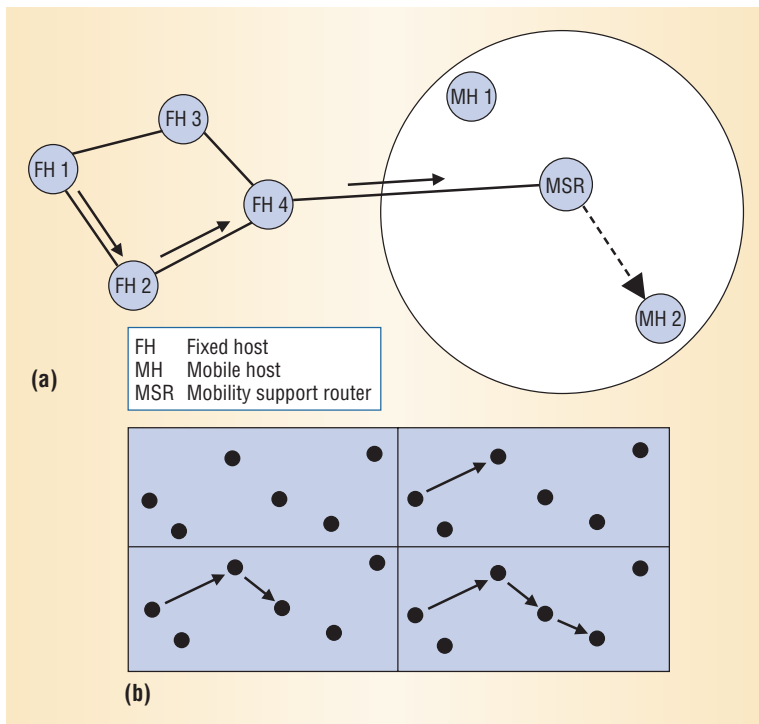


Figure 1. Routing in manets. (a) A classic cellular topology routes each packet in only one hop. (b) Manets route packets in multiple hops.

arduous environments. For example, robots or autonomous sensors deployed in an area inaccessible to humans could use simple manet routing protocols to transmit data to a control center. Even if many robots or sensors are disabled or destroyed, the remaining ones would be able to reconfigure themselves and continue transmitting information.

ROUTING IN MANETS

Efficient routing of packets is a primary manet challenge. Conventional networks typically rely on distance-vector or link-state algorithms, which depend on periodic broadcast advertisements of all routers to keep routing tables up-to-date. In some cases, manets also use these algorithms, which ensure that the route to every host is always known. However, this approach presents several problems:

- periodically updating the network topology increases bandwidth overhead;
- repeatedly awakening hosts to receive and send information quickly exhausts batteries;
- the propagation of routing information, which depends on the number of existing hosts, causes overloading, thereby reducing scalability;
- redundant routes accumulate needlessly; and
- communication systems often cannot respond to dynamic changes in the network topology quickly enough.

Manets use multihop rather than single-hop routing to deliver packets to their destination. As Figure 1a shows, a standard cellular topology routes each packet in only one hop, from the base station to the mobile host. However, as Figure 1b

shows, manets can route packets in multiple hops, enabling direct communication between mobile hosts without the need for mobility support router mediation.

ON-DEMAND ROUTING ALGORITHMS

Rather than relying on periodical broadcasts of available routes, algorithms such as dynamic source routing (DSR) and ad hoc on-demand distance vector routing (AODVR) discover routes as needed. Because the route to every mobile node is not known at any given time, these algorithms must build and maintain routes.

Dynamic source routing

DSR¹ is a fairly simple algorithm based on the concept of *source routing*, in which a sending node must provide the sequence of all nodes through which a packet will travel. Each node maintains its own *route cache*, essentially a routing table, of these addresses. Source nodes determine routes dynamically and only as needed; there are no periodic broadcasts from routers.

Figure 2 illustrates the DSR algorithm's route discovery/route reply cycle. A source node that wants to send a packet first checks its route cache. If there is a valid entry for the destination, the node sends the packet using that route; if no valid route is available in the route cache, the source node initiates the route discovery process by sending a special route request (RREQ) packet to all neighboring nodes.

The RREQ propagates through the network, collecting the addresses of all nodes visited, until it reaches the destination node or an intermediate node with a valid route to the destination node. This node in turn initiates the route reply process by sending a special route reply (RREP) packet to the originating node announcing the newly discovered route. The destination node can accomplish this using inverse routing or by initiating the route discovery process backwards.

The DSR algorithm also includes a route maintenance feature implemented via a hop-to-hop or end-to-end acknowledgment mechanism; the former includes error checking at each hop, while the latter checks for errors only on the sending and receiving sides. When the host encounters a broken link, it sends a route error (RERR) packet.

Dynamic source routing is easy to implement, can work with asymmetric links, and involves no overhead when there are no changes in the network. The protocol can also easily be improved to support multiple routes to the same destination.

DSR's main drawback is the large bandwidth

overhead inherent in source routing. Because each route cache collects the addresses of all visited nodes, RREQ packets can become huge as they propagate through the network. Routing information can also increase enough to exceed the accompanying message's usefulness. These problems limit the network's acceptable diameter and therefore its scalability.

Ad hoc on-demand distance vector routing

With AODVR,² a source node that wants to send a message to a destination for which it does not have a route broadcasts an RREQ packet across the network. All nodes receiving this packet update their information for the source node. Thus, unlike DSR, this approach does not use route caching. Instead, each node maintains only the next hop's address in a routing table, and these routing tables are updated all the way along the RREQ propagation path.

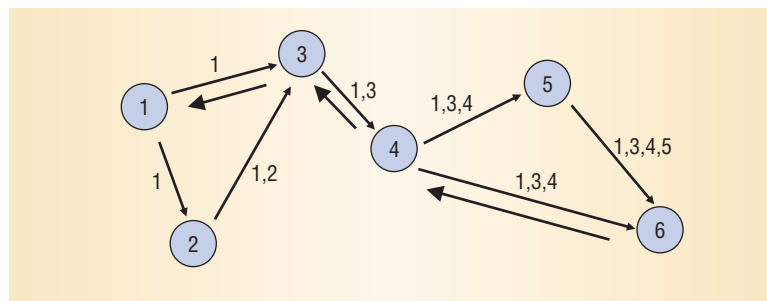
The RREQ contains the source node's address, broadcast ID, and current sequence number as well as the destination node's most recent sequence number. Nodes use these sequence numbers to detect active routes. A node that receives an RREQ can send an RREP if it either is the destination or has a route to the destination with a corresponding sequence number greater than or equal to the sequence number the RREQ contains. In the latter case, the node returns an RREP to the source with an updated sequence number for that destination; otherwise, it rebroadcasts the RREQ.

Nodes keep track of the RREQ source address and broadcast ID, discarding any RREQ they have already processed. As the RREP propagates back to the source, nodes set up entries to the destination in their routing tables. The route is established once the source node receives the RREP.

This algorithm also includes route maintenance facilities. For every route in a routing table, a host maintains a list of neighboring nodes using that route and informs them about potential link breakages with RERR messages. Each node also records individual routing table entries and deletes those not used recently.

AODVR offers several key advantages compared to DSR³:

- it supports multicast by constructing trees connecting all the multicast members along with the required nodes;
- smaller control and message packets result in less network bandwidth overhead; and
- the need for only two addresses when routing—destination and next hop—rather than the entire sequence ensures good scalability



because packet size does not depend on network diameter.

However, AODVR only works with symmetric links, and because it does not allow for multipath routing, new routes must be discovered when a link breaks down.

LINK-STATE ROUTING ALGORITHMS

Link-state routing algorithms exploit the periodic exchange of control messages between routers, ensuring that the route to every host is always known and immediately providing required routes as needed. However, this proactivity comes at the cost of high bandwidth overhead. Ad hoc link-state routing algorithms attempt to conserve bandwidth by reducing the size and number of control messages.

Optimized link-state routing

Classic link-state algorithms declare all links with neighboring nodes and flood the entire network with routing messages. Optimized link-state routing⁴ compacts control packet size by declaring only *multipoint relay selectors*, a subset of neighboring links. To further reduce traffic, OLSR uses only the selected nodes, called *multipoint relays* (MPRs), to flood the network with routing messages.

Each node selects a set of neighboring nodes as MPRs, and these nodes rebroadcast packets received from the originating node. Thus, unlike ordinary broadcast, not every node forwards routing messages. Each node maintains a table of MPR selectors and rebroadcasts every message coming from those selectors. In this way, the network distributes only partial link-state information, which OLSR can use to calculate an optimal route in terms of number of hops.

Each node periodically broadcasts hello messages containing information about its neighbors and a link status. Nodes select the minimal subset of MPRs among one-hop neighbors to cover all nodes two hops away. Thus, every node in the two-hop neighborhood must have a symmetric link to a given node's MPR set.

Because OLSR significantly reduces the number of broadcast retransmissions, this algorithm is most effective in networks with dense node distribution and frequent communication.

Figure 2. Dynamic source routing. A source node (1) sends a special route request packet to all neighboring nodes, and it propagates through the network. Upon receiving the RREQ, the destination node (6) sends a special route reply packet to the originating node announcing the newly discovered route.

A hybrid approach captures the advantages of on-demand and optimized link-state routing for wireless sensor networks.

Topology broadcast based on reverse-path forwarding

TBRPF⁵ broadcasts link-state updates via *source trees* that provide paths to all reachable nodes. It computes these source trees with partial topology information using a modification of Dijkstra's algorithm. Similar to OLSR, each node declares only part of its source tree to neighbors.

TBRPF uses both periodic broadcasts and differential updates to report updates, but each node can declare a full tree, leading to

the entire topology's link-state behavior. Each route update travels along a single path to every node on a source tree; leaves do not forward updates. Nodes discover neighbors using differential hello messages that only report changes in the neighborhood, which makes the messages smaller than those in OLSR.

This algorithm is useful in dense mobile networks. Unlike OLSR, it is not limited to two-hop trees, which eliminates redundancy while delivering routing information. Also, while OLSR computes only routes with a minimum number of hops, TBRPF can use arbitrary link metrics if the links are symmetric.

HYBRID APPROACH

A recently proposed hybrid approach⁶ captures the advantages of on-demand and optimized link-state routing for wireless sensor networks. This algorithm discovers the route to each node only when it is needed. However, route discovery does not occur through simple flooding but through a mechanism similar to multipoint relays.

The algorithm defines three types of nodes: master, gateway, and plain. A group of nodes selects a master to form a piconet and then synchronizes and maintains the neighbor list. A node can be a master in only one piconet, but it can be a plain member in any number of piconets. Gateway nodes belong to two or more piconets. Only masters and gateways forward routing information; plain nodes receive and process this information, but they do not forward it.

Simulation shows that this algorithm works best when the piconets are densely populated; otherwise, it degrades to simple network flooding. Future research should focus on using some well-defined and accepted metrics, such as power consumption, to compare various ad hoc routing approaches.⁷

SECURITY IN MANETS

The use of wireless links makes manets susceptible to attack. Eavesdroppers can access secret infor-

mation, violating network confidentiality. Hackers can directly attack the network to delete messages, inject erroneous messages, or impersonate a node, which violates availability, integrity, authentication, and nonrepudiation. Compromised nodes also can launch attacks from within a network.

On-demand and link-state routing algorithms do not specify a scheme to protect data or sensitive routing information. Because any centralized entity could lead to significant vulnerability in manets, a security solution must be based on the principle of distributed trust.

This is similar to the dilemma posed by the classic Byzantine generals problem,⁸ in which a general commands each division of the army, and some of the generals, who communicate via messenger, are traitors. All loyal generals must decide upon the same plan of action—that is, a small number of traitors cannot cause the loyal generals to adopt a bad plan. The same holds for manets: A number of compromised nodes cannot cause the network to fail.

Although no single node in a manet is trustworthy, threshold cryptography can distribute trust to an aggregation of nodes.⁹ This scheme lets n parties share the ability to perform a cryptographic operation such that any t parties can do it together, while up to $t - 1$ parties cannot perform the operation. However, dividing a private key into n shares and constructing t partial signatures is nontrivial given that traditional key distribution schemes either do not apply to the ad hoc scenario or are not efficient for resource-constrained devices.

Combining identity-based techniques with threshold cryptography can achieve flexible and efficient key distribution.¹⁰ After distribution, a combiner can verify the t signatures and compute the final signature for the certificate. In this way, up to $t - 1$ compromised nodes cannot generate a valid certificate by themselves.

If a large number of nodes are compromised, attributing fault to a specific malicious node is impossible. A proposed algorithm¹¹ addresses this problem by limiting the possible fault location to the link between two adjacent nodes; as long as a fault-free path exists between two nodes, they can establish a secure communication link even if most nodes in the network are compromised. In addition, this algorithm can detect selfish nodes that refuse to cooperate with other nodes. If their behavior is the result of a denial-of-service attack rather than power-savings activity, the algorithm can isolate the selfish nodes.

Wireless research today primarily focuses on the functional aspect of manets—improving the delivery of packets from one node to another. However, as technology matures, non-functional properties such as semantics and security will play the leading role. The challenge lies in managing these two layers, which are orthogonal to each other. If ad hoc communication is to be the foundation for pervasive computing, we must be able to seamlessly interconnect different platforms and devices, offer services on demand, and make it all secure and trusted. ■

References

1. D.B. Johnson, D.A. Maltz, and Y-C. Hu, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)," IETF Mobile Ad Hoc Networks Working Group, Internet Draft, work in progress, 15 Apr. 2003.
2. C.E. Perkins, E.M. Belding-Royer, and S.R. Das, "Ad Hoc On-Demand Distance Vector (AODV) Routing," IETF Mobile Ad Hoc Networks Working Group, Internet Draft, work in progress, 17 Feb. 2003.
3. S.R. Das, C.E. Perkins, and E.M. Belding-Royer, "Performance Comparison of Two On-Demand Routing Protocols for Ad Hoc Networks," *Proc. IEEE Infocom 2000*, vol. 1, IEEE Press, 2000, pp. 3-12.
4. P. Jacquet et al., "Optimized Link State Routing Protocol for Ad Hoc Networks," *Proc. IEEE Int'l Multi Topic Conf., 2001*, IEEE Press, 2001, pp. 62-68.
5. R. Ogier, F. Templin, and M. Lewis, "Topology Dissemination Based on Reverse-Path Forwarding (TBRPF)," IETF Mobile Ad Hoc Networks Working Group, Internet Draft, work in progress, 14 Oct. 2003.
6. N. Milanovic et al., "Bluetooth Ad-Hoc Sensor Network," *Proc. 2002 Int'l Conf. Advances in Infrastructure for e-Business, e-Education, e-Science, and e-Medicine on the Internet*, Scuola Superiore G. Reiss Romoli, 2002; www.informatik.hu-berlin.de/~milanovi/bt_adhoc_sensor.pdf.
7. I. Stojmenovic and X. Lin, "Power-Aware Localized Routing in Wireless Networks," *IEEE Trans. Parallel and Distributed Systems*, vol. 12, no. 11, 2001, pp. 1122-1133.
8. L. Lamport, R.E. Shostak, and M. Pease, "The Byzantine Generals Problem," *ACM Trans. Programming Languages and Systems*, vol. 4, no. 3, 1982, pp. 382-401.
9. Y. Desmedt, "Some Recent Research Aspects of Threshold Cryptography," *Proc. 1st Ann. Workshop Information Security*, LNCS 1396, Springer-Verlag, 1997, pp. 158-173.
10. A. Khalili, J. Katz, and W.A. Arbaugh, "Toward Secure Key Distribution in Truly Ad-Hoc Networks," *2003 Symp. Applications and the Internet Workshops (SAINT 03 Workshops)*, IEEE CS Press, 2003, pp. 342-346.
11. B. Awerbuch et al., "An On-Demand Secure Routing Protocol Resilient to Byzantine Failures," *Proc. ACM Workshop Wireless Security*, ACM Press, 2002, pp. 21-30.

Nikola Milanovic is a PhD student at the Institute for Informatics, Humboldt University, Berlin. His research interests include wireless communications, ad hoc networking, ubiquitous computing, and component- and service-based environments. Milanovic received a Dipl.-Ing. in electrical engineering from the University of Belgrade. He is a member of the IEEE. Contact him at milanovi@informatik.hu-berlin.de.

Mirosław Malek is a professor and chair of computer architecture and communication at Humboldt University. His research focus is on high-performance responsive computing, including parallel architectures, real-time systems, networks, and fault tolerance. Malek received a PhD in computer science from the Technical University of Wrocław, Poland. He is a member of the ACM. Contact him at malek@informatik.hu-berlin.de.

Anthony Davidson is a professor and director of graduate programs in management and systems at New York University's School of Continuing and Professional Studies. His research interests include systems science, enterprise management, strategic planning, and operations. Davidson received a PhD in management and systems science from the City University of London. He is a member of the Association for the Advancement of Computing in Education. Contact him at anthony.davidson@nyu.edu.

Veljko Milutinovic is a professor in the Department of Computer Engineering at the University of Belgrade School of Electrical Engineering. His research interests include infrastructure for e-business on the Internet, shared memory multiprocessors, distributed shared memory, and microprocessor architecture and design. Milutinovic received a PhD in electrical engineering from the University of Belgrade. He is a Fellow of the IEEE. Contact him at vm@etf.bg.ac.yu.