

ROUTING IN INTERMITTENTLY CONNECTED MOBILE AD HOC NETWORKS AND DELAY TOLERANT NETWORKS: OVERVIEW AND CHALLENGES

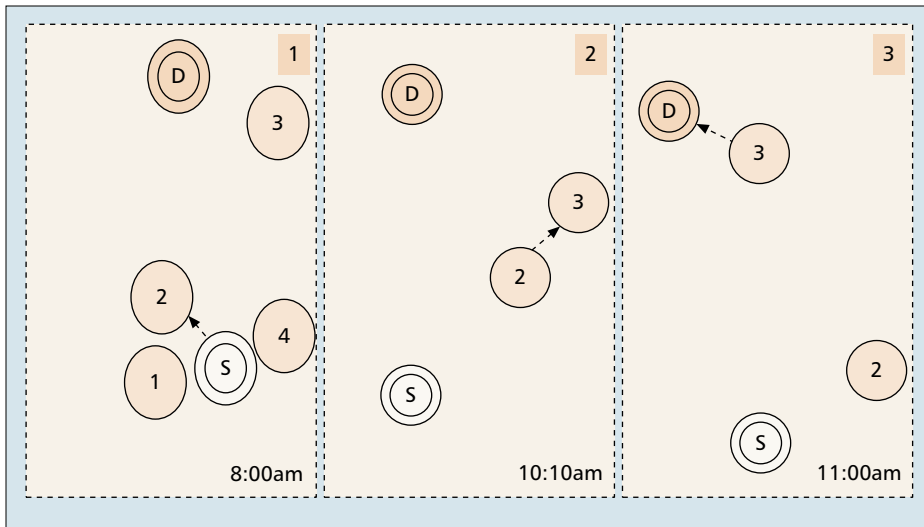
ZHENSHENG ZHANG, SAN DIEGO RESEARCH CENTER

ABSTRACT

Recently there has been much research activity in the emerging area of intermittently connected ad hoc networks and delay/disruption tolerant networks (DTN). There are different types of DTNs, depending on the nature of the network environment. Routing in DTNs is one of the key components in the DTN architecture. Therefore, in the last few years researchers have proposed different routing protocols for different types of DTNs. In this article we capture the state of the art in routing protocols in DTNs. We categorize these routing protocols based on information used. For deterministic time evolving networks, three main approaches are discussed: the tree approach, the space and time approach, and the modified shortest path approach. For stochastic time evolving networks, the following approaches are discussed: the epidemic or random forwarding approach, predication or history based approach (including per contact routing based on one hop information only and per contact routing based on average end to end information), the model based routing approach, as well as approaches that control the movement of certain special nodes. Recent developments in erasure coding and network coding applied to DTNs are also discussed. The article also identifies open research issues and intends to motivate new research and development in this area.

In the last few years, there has been much research activity in mobile, wireless, ad hoc networks (MANET). MANETs are infrastructure-less, and nodes in the networks are constantly moving. In MANETs, nodes can directly communicate with each other if they enter each others' communication range. A node can terminate packets or forward packets (serve as a relay). Thus, a packet traverses an ad hoc network by being relayed from one node to another, until it reaches its destination. As nodes are moving, this becomes a challenging task, since the topology of the network is in constant change. How to find a destination, how to route to that destination, and how to insure robust communication in the face of constant topology change are major challenges in mobile ad hoc networks. Routing in mobile ad hoc networks is a well-studied topic. To accommodate the dynamic topology of mobile ad hoc networks, an abundance of routing protocols have recent-

ly been proposed, such as OLSR [1], AODV [2], DSR [3], LAR [4], EASE [5, 6], ODMRP [7], and many others [8, 9]. For all these routing protocols, it is implicitly assumed that the network is connected and there is a contemporaneous end-to-end path between any source and destination pair. However, in a physical ad hoc network, the assumption that there is a contemporaneous end-to-end path between any source and destination pair may not be true, as illustrated below. In MANETs, when nodes are in motion, links can be obstructed by intervening objects. When nodes must conserve power, links are shut down periodically. These events result in intermittent connectivity. At any given time, when no path exists between source and destination, network partition is said to occur. Thus, it is perfectly possible that two nodes may never be part of the same connected portion of the network. Figure 1 illustrates the time evolving behavior in intermittent-



■ **Figure 1.** Illustration of time evolving behavior of ad hoc networks.

ly connected networks (ICN). In Fig. 1 there is no direct path from node S to node D at any given time. Packets from node S can be delivered to node D if intermediate nodes can hold/carry the packets. (At 8:00 am, node S sends the packets to node 2; at 10:00am node 2 forwards the packets to node 3; and at 11:30 am node 3 forwards the packets to node D.) Examples of an intermittently connected network are:

- An inter-planet satellite communication network where satellites and ground nodes may only communicate with each other several times a day.
- A sensor network where sensors are not powerful enough to send data to a collecting server all the time or scheduled to be wake/sleep periodically.
- A military ad hoc network where nodes (e.g., tanks, airplanes, soldiers) may move randomly and are subject to being destroyed.

Applications in ICNs must tolerate delays beyond conventional IP forwarding delays, and these networks are referred to as delay/disruption tolerant networks (DTN). *Routing protocols such AODV and OLSR do not work properly in DTNs, since under these protocols, when packets arrive and no contemporaneous end-to-end paths for their destinations can be found, these packets are simply dropped. New routing protocols and system architectures should be developed for DTNs.*

There are many potential applications in DTNs, such as inter-planetary network (IPN), Zebranet, DataMule, and village networks. IPN [10] consists of both terrestrial and inter-planetary links, which suffers from long delays and episodic connectivity. In Zebranet [11], wild-life researchers drive through a forest collecting information about the dispersed zebra population. In the DataMule project [12], DataMules randomly move and collect data from low power sensors. For village networks, for example, a recent project in developing nations uses rural buses to provide Internet connectivity to otherwise isolated and remote villages that do not have any communication infrastructure [13]. Another example of village networks is presented in [14], in which the Wizzy digital courier service provides disconnected Internet access to people/students in remote villages of South Africa. A courier on a motorbike, equipped with a USB storage device, travels from a village to a large city that has high-speed Internet connectivity. Typically, it takes a few hours for the courier to travel from the village to the city.

There are many different terminologies used for DTNs in the literature, such as eventual connectivity, space-time routing, partially connected, transient connection, opportunistic networking, extreme networks, and end-to-end communi-

tion. The data unit in DTNs can be a message, a packet, or bundle, which is defined as a number of messages to be delivered together. For simplicity, throughout this article we use bundles, messages, and packets interchangeably.

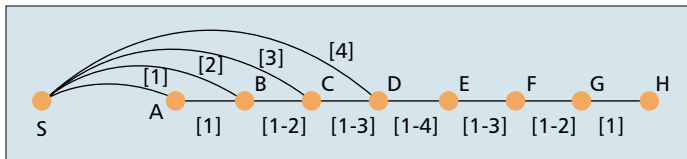
The characteristics of DTNs are very different from the traditional Internet in that the latter implicitly has some well-known assumptions:

- Continuous connectivity
- Very low packet loss rate
- Reasonably low propagation delay

DTNs do not satisfy all of these assumptions, and sometimes none. The challenges in designing efficient protocols in DTNs are extremely long delay (up to days),

frequent disconnection, and opportunistic or predicable connections. Consequently, the existing protocols developed for the wired Internet are not able to handle data transmission efficiently in DTNs. In DTNs, end-to-end communication using the TCP/IP protocol may not work, as packets whose destinations cannot be found are usually dropped. If packet dropping is too severe, TCP eventually ends the session. UDP provides no reliable service and cannot “hold and forward.” *New protocols and algorithms need to be developed.* There are several different types of DTNs due to their different characteristics. For instance, the satellite trajectories in example a) are predictable, while the movement of a soldier or tank in example c) may be random. Therefore, for different types of DTNs, different solutions may need to be proposed. Recently, the DTN research group under the Internet Research Task Force (IRTF) has proposed several research documents, including a DTN architecture [10, 15–17]. This architecture addresses communication issues in extreme networks or networks encompassing a wide range of architectures. The architecture is a network of regional networks, and an overlay on top of the transport layer of these regional networks. It provides key services, such as in-network data storage and retransmission, interoperable naming, and authenticated forwarding. The DTN solutions in [15] are concerned with message transport between infrastructures of disparate architectures by using gateways that handle “bundles” of messages between these infrastructures. The DTN architecture addresses the issues of eventual connectivity and partitioned networks by the use of a store and forward mechanism, and handles the diverse addressing needs of the overlay architecture by using an addressing scheme that exploits the late binding of addresses — local addresses are not bound to nodes until the message is in the local area of the destination. This creates a hierarchical routing structure that makes routing across networks easier to implement. There are many activities in the DTN working group; due to space limitation, readers are referred to [15] (the references therein) for more information.

Routing in DTNs is one of the key components in the architecture document. Based on different types of DTNs, deterministic or stochastic, different routing protocols are required. Due to intermittent connectivity, it is likely that paths to some of the destinations may not exist from time to time. When a packet arrives and its destination cannot be found in the routing table, the packet is simply dropped under the routing protocols mentioned above (developed with the assumption that the network is connected). Therefore, these



■ **Figure 2.** Illustration of different metrics.

routing protocols will not work efficiently in DTNs. In this article, we provide an overview of the state of the art in DTN routing protocols.

To cope with intermittent connectivity, one natural approach is to extend the store-and-forward routing to store-carry-forward (SCF) routing. In store-carry-forward routing, a next hop may not be immediately available for the current node to forward the data. The node will need to buffer the data until the node gets an opportunity to forward the data and must be capable of buffering the data for a considerable duration. The difficulty in designing a protocol for efficiently and successfully delivering messages to their destinations is to determine, for each message, the best nodes and time to forward. If a message cannot be delivered immediately due to network partition, the best carriers for a message are those that have the highest chance of successful delivery, i.e., the highest delivery probabilities. As ad hoc networks could be very sparse, SCF routing could mean that the node may have to buffer data for a long period of time. This condition can become worse if the next hop is not selected properly. A bad forwarding decision may cause the packets to be delayed indefinitely. If messages must be stored somewhere, a buffer management scheme should be proposed.

If all the future topology of the network (as a time-evolving graph) is deterministic and known, or at least predictable, the transmission (when and where to forward packets) can be scheduled ahead of time so that some optimal objective can be achieved. If the time-evolving topology is stochastic, SCF routing performs routing by moving the message closer to the destination one hop at a time. If the nodes know nothing about the network states, then all the nodes can do is to randomly forward packets to their neighbors. Protocols in this category are referred to as epidemic. If one can estimate the forwarding probability of its neighbors, a better decision could be made. Protocols in this category are referred to as history-based or estimation-based forwarding. Furthermore, if the mobility patterns can be used in the forwarding probability estimation, an even better decision may be made. Protocols in this category are referred to as model-based forwarding. In some cases, network efficiency can be achieved if the movements of certain nodes are controlled and these protocols are in the category of controlling node movements. Recently, coding-based routing protocols have also been proposed for DTNs.

In this article we will review some of the routing protocols in DTNs and categorize them as follows.

- Deterministic case
 - Space time routing [18]
 - Tree approach [19]
 - Modified shortest path approaches [16]
 - Stochastic case
 - Epidemic/random spray [20–25]
 - History or predication-based approach
 - ✓ Per contact routing based on one-hop information [11, 26–29]
 - ✓ Per contact routing based on end-to-end information [26, 30, 31]
 - Model-based [32, 33]
 - Control movement [12, 34–39]
 - Coding-based approaches [40–43]
- The remainder of the article is organized as follows. We

review protocols under deterministic assumption (where the network topology is deterministic and known ahead of time). Protocols for stochastic or dynamic networks are discussed, including recent developments using coding techniques. Future research topics and challenges are discussed. We then conclude the article.

DETERMINISTIC ROUTING

In this section we review a few routing protocols, assuming that future movement and connections are completely known (that is, the entire network topology is known ahead of time).

In a tutorial article [44], Ferreira describes a simple combinatorial reference model that captures most characteristics of time-varying networks. A notion of evolving graphs, which consists of formalizing time domain graphs, is introduced. Modeling time in mobile ad hoc networks gives rise to several different matrices that may serve as objective functions in routing strategies, such as “earliest time to reach one or all the destinations” or “minimum hop paths” (with or without the condition that packets arrive before a predefined time period). Readers are referred to [44] for more details and references therein. In Fig. 2 of [44] the min-hop path from S to D takes four hops at time interval 1, while it takes one hop at time interval 4, where the number next to each link denotes the time interval during which the link is active.

In [19] algorithms selecting the path of message delivering are presented, depending on the available knowledge about the motion of hosts. Three cases are considered. In the first case it assumes that global knowledge of the characteristic profiles with respect to space and time (that is, the characteristic profiles of the motion and availability of the hosts as *functions of time*) are completely known by all the hosts. Paths are selected by building a tree first. Such an approach is referred to as the tree approach. Under the tree approach, a tree is built from the source host by adding children nodes and the time associated with nodes. Each node records all the previous nodes the message has to travel and the earliest time to reach it. A final path can be selected from the tree by choosing the earliest time (or minimum hop) to reach the desired destination. In the second case, it assumes that characteristic profiles are initially unknown to hosts. Hosts gain this information through learning the future by letting neighbor hosts exchange the characteristic profiles available between them. Paths are selected based on this partial knowledge. In the third case, to enhance the algorithm in the second case, it also requires hosts to record the past, that is, it stores the sequence of hosts a message has transited within the message itself.

For DTNs, several routing algorithms are proposed in [16] depending on the amount of knowledge about the network topology characteristics and traffic demand. They define four *knowledge oracles*; each oracle represents certain knowledge of the network. The *Contacts Summary Oracle* contains information about aggregate statistics of the contacts (resulting in time-invariant information). A contact is defined as an opportunity to send data. The *Contacts Oracle* contains information about contacts between two nodes at any point in time. This is equivalent to knowing the time-varying networks. The *Queueing Oracle* gives information about instantaneous buffer occupancies (queueing) at any node at any time. The *Traffic Demand Oracle* contains information about the present or future traffic demand. Based on the assumption of which oracles are available, the authors present corresponding routing algorithms. For example, if all the oracles are known, a linear

programming is formulated to find the best route. If only the Contacts Summary Oracle is available, Dijkstra with time-invariant edge costs based on average waiting time is used to find the best route. If only the Contact Oracle is available, modified Dijkstra with time-varying cost function based on waiting time is used to find the route. All of the algorithms developed (except the zero-knowledge case) in [16] are for the deterministic case.

Assuming the characteristic profile is known over an infinite time horizon may not be realistic in ad hoc networks. In [18] it is assumed that the characteristic profile can be accurately predicted over the time interval of T . They model the dynamic of the networks as a *space-time graph*. Routing algorithms in the constructed space-time graph are developed using dynamic programming and shortest path algorithm. The routing algorithm finds the best route for messages by looking ahead. The idea of the time layers in the space-time graph comes from time-expanded graphs [45]. The time-expanded graphs approach translates a problem of network flow over time to a classical “static” network flow, and standard tools of graph theory such as the Floyd-Warshall algorithm can be applied to compute the shortest path for a source destination pair.

In all these approaches under the deterministic case, an end-to-end path (possibly time dependent) is determined before messages are actually transmitted. However, in certain cases the topology of the network may not be known ahead of time. In the following section we review some of the protocols designed for stochastic or random networks.

STOCHASTIC OR DYNAMIC NETWORKS

In this section, we review some of the routing protocols when the network behavior is random and not known. These protocols depend on decisions regarding where and when to forward messages. The simplest decision is to forward to any contacts within range, while other decisions are based on history data, mobility patterns, or other information.

EPIDEMIC (OR PARTIAL) ROUTING-BASED APPROACH

In the epidemic routing category, packets received at intermediate nodes are forwarded to all or part of the nodes’ neighbors (except the one who sends the packet) without using any predication of the link or path forwarding probability. Epidemic routing is a natural approach when no information can be determined about the movement patterns of nodes in the system.

Vahdat and Becker [20] propose an epidemic routing protocol for intermittently connected networks. When a message arrives at an intermediate node, the node floods the message to all its neighbors. Hence, messages are quickly distributed through the connected portions of the network. Epidemic routing relies on carriers of messages coming into contact with another node through node mobility. When two nodes are within communication range, they exchange pair-wise messages that the other node has not seen yet. Their simulation results show that, in the special scenarios considered, epidemic routing is able to deliver nearly all transmitted messages, while existing ad hoc routing protocols fail to deliver any messages because of the limited node connectivity when the buffer capacity is sufficiently large.

Another extreme is to let the source hold the message and deliver to the destination only when they are within communication range. This approach obviously has minimal overhead, but the delay could be very long. In [21] Grossglauser and Tse

propose a 2-hop forwarding approach and have explored a theoretical framework where nodes with infinite buffer move independently around the network, and every node gets close to any other node for some short time period per time slot. Within this framework, a node s gives a message addressed to node t to another *randomly* chosen node one hop away in the network, called a “receiver.” When the receiver happens to be within the range of the destination node t , the receiver sends the message to the destination. Hence, a message will only make *two hops* and no message will be transmitted more than twice. They prove that a message is guaranteed to be delivered, even if its delivery time is averaged over many time slots. This result sets a theoretical bound, since it assumes a complete mixing of the trajectories so that every node can get close to another one.

In the Infostation model [46] users can connect to the network in the vicinity of Infostations, which are geographically distributed throughout the area of network coverage. Infostations provide strong radio signal quality to small disjoint geographical areas and, as a result, offer very high rates to users in these areas. However, due to the lack of continuous coverage, this high data rate comes at the expense of providing intermittent connectivity only. Since a node that wishes to transmit data may be located outside the Infostations’ coverage areas for an extended period of time and must always transmit to an Infostation directly, large delays may result. Upon arrival in a coverage zone, the node can transmit at very high bit-rates. Thus, *Infostations trade connectivity for capacity, by exploiting the mobility of the nodes*. It is assumed that the *Infostations* are connected. The authors in [22] propose a Shared Wireless Infostation Model (SWIM), where SWIM is a marriage of the Infostations concept with the (epidemic) ad hoc networking model. (Propagation of information packets within SWIM is identical to the epidemic routing protocol [20].) The only difference is that any one of the many Infostations could serve as a destination node, while in [20], there is only one destination node for a given packet. A real-world application based on the Infostation model is presented in [22]. One of the benefits of SWIM, by allowing the packet to spread throughout the mobile nodes, is that the delay for the replicas to reach an Infostation can be significantly reduced. However, this comes at a price: spreading the packets to other nodes consumes network capacity. Again, there is a capacity-delay tradeoff, which can be controlled by limiting the parameters of the spread, for example, by controlling the probability of packet transmission between two adjacent nodes, the transmission range of each node, or the number and distribution of the Infostations. However, how to choose these controlling parameters is not discussed in [22].

A relay-based approach to be used in conjunction with traditional ad hoc routing protocols is proposed in [23]. This approach takes advantage of node mobility to disseminate messages to mobile nodes. The result is the Mobile Relay Protocol (MRP), which *integrates message routing and storage in the network*. The basic idea is that if a route to a destination is unavailable, a node performs a *controlled local broadcast* (a relay) to its immediate neighbors (that is the only time that broadcast is used in the protocol). All nodes that receive this packet store it and enter the relaying mode. In the relaying mode, the MRP first checks with the (traditional) routing protocols to see if a route of less than d hops exists to forward the packet. If so, it forwards the packet and the packet is delivered. If no valid route exists for the packet, it enters the storage phase, which consists of the following steps:

- If the packet is already stored in the node’s buffer, then the older version of the packet is discarded.

- Otherwise, the node buffer is checked. If it is not full, then the packet is stored and the time-to-live parameter h in the MRP header of the packet is decremented by 1.
- If the buffer is full, then the least recent packet is removed from the buffer and it is relayed to a single *random* neighbor if $h > 0$.

In a network with sufficient mobility, it is quite likely that one of the relay nodes to which the packet has been relayed will encounter a *node* that has a valid, short (conventional) route to the eventual destination, thereby increasing the likelihood that the message will be successfully delivered. It is not clear why the authors choose to broadcast only once, and then select one node *randomly* to forward in case the buffer is full (to keep the packet in the network). Why not broadcast twice or three times?

To limit the amount of broadcasting to all its neighbors as in [20, 22], the Spraying protocol [24] restricts forwarding to a ray in the vicinity of the destination's last known location. In [24] it is assumed that the destination's *last location is known* and there is a separate *location manager* in the system. To deal with high mobility, Spray routing [24] multicasts traffic within the vicinity of the last known location of a session's destination. The idea is that even though a highly mobile node may not be in the location last reported by the location tracking mechanism, it is likely to be in one of the surrounding locations. By "spraying" to the vicinity of the *last-known location* of the destination, the algorithm attempts to deliver packets to the destination even if it moves to a nearby location during the location tracking convergence time. A sprayed packet is first unicast to a node close to the destination, and then multicast to multiple nodes around the destination. The magnitude of the spraying depends on the mobility; the higher the mobility, the larger the vicinity. Upon a change in affiliation, a node sends a *location update* to its location manager. In order to communicate with a destination node, D, a source node sends a *location subscribe* to the location manager. The current location and changes (in location) thereafter are sent by the location manager to the source using a location information message. Note that it is possible that the destination receives duplicate packets, and it is assumed that there is an end-to-end duplicate detection mechanism that will discard such packets. Spray routing is an integrated location tracking and forwarding scheme. Both location managers and switches/routers participate in spray routing. How to choose the first node at the beginning to forward to is not clearly explained in the article.

In [25] the authors present an experimental study to test the feasibility of using user mobility and opportunistic pair-wise contact to form an *opportunistic ad-hoc* network in a *campus environment*. Using commodity mobile devices, they instrument two user studies for experimentally collecting trace data of user contact. The approach is unique in that they do not have a predetermined model of user mobility, and they strive to provide a networking model based only on pair-wise contact. The results of the experiment are promising, showing that user mobility can potentially be used to form a network. Using this trace data, they simulate an idealized network using *epidemic* propagation, and observe that nodes exhibit signs of regularity and affinity of contact. Furthermore, in many cases, success of message delivery from any source to a destination is not evenly distributed among the intermediate nodes. Thus, source nodes can potentially use this information for better routing decisions. During the test, they observe that power management is one of the critical issues. No new protocols are proposed in the article. Analysis of the delay in epidemic routing is presented in [47, 48].

ESTIMATION (OF THE LINK FORWARDING PROBABILITY) BASED APPROACH

Instead of blindly forwarding packets to all or some neighbors, intermediate nodes estimate the chance, for each outgoing link, of eventually reaching the destination. Based on this estimation, the intermediate nodes decide whether to store the packet and wait for a better chance, or decide to which nodes (and the time) to forward. A first theoretic work on link estimation is given in [49]. Some protocols make the per contact forwarding decisions based only on the next hop information, such as next hop forwarding probability, while some other protocols make the per contact forwarding decisions based on average end-to-end metrics, such as expected short-cut path or average end-to-end delay.

Per Contact Routing Based on Next Hop Information

Only — A follow up work to [20] is presented in [26], which extends Vahdat's work to situations with limited resources. In their work, though still using flooding-like propagation, they enhance the drop strategy in epidemic routing when caches or buffers are filled. Their algorithm works as follows. When a node A meets another node B, they perform a bundle exchange through a number of steps. First, node A gives to node B a list of the bundles node A carries with their destinations. Each bundle also contains a likelihood of delivery by node A. Node A receives the same list from B and calculates the likelihood of delivering B's bundles. Node A now sorts the combined lists by the likelihood of delivery, removes node A's own bundles, and also deletes bundles that B has a higher likelihood of delivering. Node A then selects the top n bundles remaining, and *requests from B all the bundles (up to n) that node A does not already have*. They propose four types of drop strategies for deciding which bundles to exchange when two nodes meet, and simulation results show that Drop-Oldest (DOA) and Drop-Least-Encountered (DLE) yielded the best performance. The encounter value is an estimation of meeting likelihood of two given nodes. The DLE algorithm has peers keep track of the other peers they meet regularly over time. Peers initialize their likelihood of delivery of a bundle to a moving peer as 0. When peer A meets another peer B, the former sets the likelihood of delivering bundles to B as 1. These values degrade over time, such that they are reinforced only if A and B meet periodically.

Similar to the work in [26], a probabilistic routing protocol, PROPHET (Probabilistic Routing Protocol using History of Encounters and Transitivity), is proposed in [27]. PROPHET first estimates a probabilistic metric called delivery predictability, $P(a, b)$, at every node a , for each known destination b . This indicates how likely it is that this node will be able to deliver a message to that destination. The operation of PROPHET is similar to that in [26]. When two nodes meet, they exchange summary vectors, and also a delivery predictability vector containing the delivery predictability information for destinations known by the nodes. The summary vectors are obtained in the same way as in [26] (where the vector is called a list). This additional information is used to update the internal delivery predictability vector as follows:

$$P_{(a,b)} = \begin{cases} P_{(a,b)_{old}} + (1 - P_{(a,b)_{old}})P_{init} & \text{if } a \text{ } b \text{ meets} \\ P_{(a,b)_{old}} \gamma^k & \text{otherwise} \end{cases}$$

where P_{init} in $(0,1)$ is an initialization constant (with all $P(a, b)$ being set at P_{init}) and γ in $(0,1)$ is an aging constant. The information in the summary vector is used to decide which messages to request from the other node. Simulation results

show that for the network considered, the improvement of packet delivery ratio under PROPHET over the epidemic routing can be up to 40 percent.

Similar to PROPHET, the Context-Aware Routing (CAR) protocol [28] integrates synchronous and asynchronous mechanisms for message delivery. Here synchronous delivery means that at the time a packet arrives, a path to its destination exists, and the packet can be forwarded using an existing routing protocol. By asynchronous delivery, at the time a packet arrives, a path to its destination cannot be found and the packet has to be stored somewhere waiting to be forwarded. If the synchronous delivery of the message is not possible since the receiver is not in the same connected portion of the mobile network, instead of replicating the message to all the neighbors, the message is sent to a host characterized by the highest probability of reaching the recipient. In other words, this host acts as a message carrier. This process is based on the evaluation and on the prediction of the context information using a time series analysis technique, Kalman Filter. Delivery probabilities are synthesized locally from context information such as the rate of change of connectivity of a host. The prediction process is used during temporary disconnections and it is carried out until it is possible to guarantee with certain accuracy. In this work it is assumed that a proactive routing protocol, such as DSDV [50], is used within a connected portion of the entire network. Simulation results indicate that when the buffer capacity is small, the packet delivery ratio of CAR is better than Epidemic routing, while the opposite is true when the buffer capacity is large. Note that there is only a single copy of the packet inside the network under CAR.

In [29] the authors study a relay-based routing scheme for ad hoc *satellite* networks where nodes are required to buffer data for a certain period of time until the node gets an opportunity to forward it. They propose Interrogation-Based Relay Routing (IBRR), where the nodes interrogate each other to learn more about network topology and nodal capacity to make intelligent routing decisions. The main issue in interrogation-based relay routing is the next-hop selection process. Given the dynamic topology and heterogeneity of an ad hoc satellite constellation, it may be difficult to decide whether or not to forward the data to a given node. To make effective routing decisions, satellites are expected to track, to their best extent, the positions of neighboring nodes and even some distant nodes. Moreover, a satellite encountered at one moment by a node may not be the best candidate for node A to forward the data to, but it may be the only choice in the foreseeable future. In such a case, node A has to decide whether to forward the data or bet its luck and wait for the next opportunity. Optimistic forwarding proposed in [33] (discussed later) is used in IBRR. To select the next hop, a node needs to not only know the present and future connectivity relation with its current time, but also the same information of its current neighbors. This one-hop “look-ahead” is necessary for making routing decisions in the relay-based routing framework. Look-ahead beyond one-hop can prove to be time consuming and counter productive. To select the next hop, a node needs to evaluate the potential candidates and select the most promising forwarding node and forward the data to that node. The selection is based on the following metrics:

- Spatial location and orbital information of the candidate nodes.
- Bandwidth of the inter-satellite link to the candidate nodes.
- Relative velocity/mobility between two nodes.
- Vicinity of this candidate to other satellites and ground stations.

- Capability of the candidate satellites.
- Data transmission time.

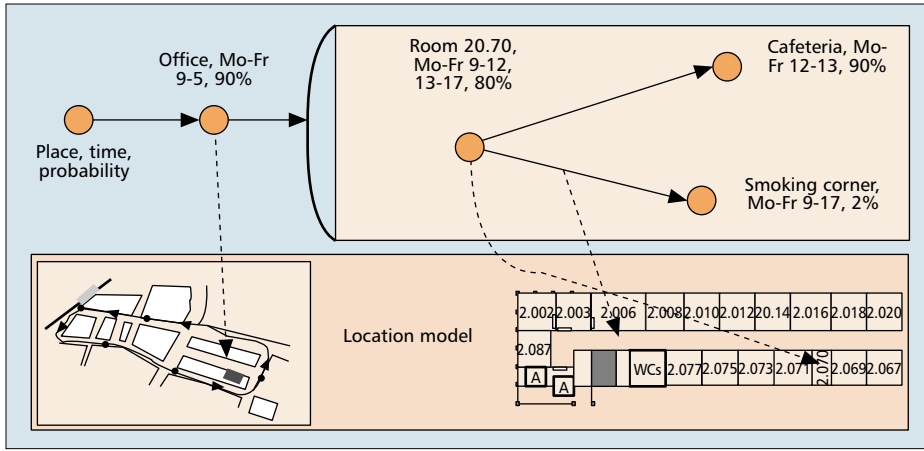
The authors propose to use interrogation where the nodes do not transmit any hello messages except to initialize a session between two neighboring nodes. After the initialization, the IBRR protocol proceeds to exchange orbital and routing information between nodes in the form of *queries and responses*, and may or may not continue with actual data transmission. Instead of discovering the entire path to the destination, a node puts more effort into acquiring information about the immediate neighbors and that of the neighbors’ neighbors (one-hop look-ahead). The *best next-hop candidates for node A* are nodes that have sent out the most number of replies to node A’s beacons. In the simulations, the authors choose the number of best next-hops to forward for each node to be 3 without giving any reason.

In ZebraNet [11], wireless sensor nodes, namely collars (attached to Zebras), collect location data and opportunistically report their histories when they come in radio range of base stations, or the researchers or data collection objects, which periodically drive through (or fly-over) with receivers to collect data. Collars operate on batteries with/without solar recharge. The goal is to study the animal behaviors through designing a collar and communication protocols that work on Zebras (high data collection rate). They study two routing protocols: flood-based routing protocols and history-based protocols. In the flood-based routing protocol, data is flooded to their neighbors whenever they meet. It is expected that as nodes move extensively and meet a number of neighbors, given enough time, data will eventually reach to the base station. In the history-based routing protocol, each node is assigned a likelihood of transferring data to the base station based on its past success. A higher value corresponds to a higher probability of eventually being within the range of the base station. Data is forwarded to its neighbor with the highest transferring probability. Experimental results indicate that the flood-based protocol yields higher system throughput if the buffer capacity at each node is large enough, but the energy consumed by the flood-based protocol can be eight times that of the history-based protocol. There is a tradeoff between throughput and energy consumption. Their conclusion is that while flooding makes sense at low-radio-range and low-connectivity points in the design space, it is not a good choice in a high-connectivity regime.

Recently the authors in [51] presented simple analysis on direct delivery and flooding algorithms using queue theory. Based on the analysis, they propose a data delivery scheme for fault tolerant sensor networks. The scheme consists of two parts. The first part makes decisions on when and where to transmit data messages based on the delivery probability, which reflects the likelihood that a sensor can deliver data messages to the sink. The second part decides which messages to transmit or drop based on the fault tolerance, which indicates the importance of the messages.

Per Contact Routing Based on Average End-to-End Performance Metrics — In the protocols reviewed in the previous section, decisions about forwarding packets (when and where) are based on the likelihood of the delivery of each neighbor. No end-to-end performance is considered. In this section, we review three protocols in which decisions are based on end-to-end performance: one on the probability to deliver to the destination; one on the expected shortest path to the destination; and one on the average end-to-end delay.

Extending their previous work in [26], the meets and visits (MV) protocol [35] uses the same exchange scheme as in [26], but presents a new method to estimate the likelihood of for-



■ **Figure 3.** Local model and user profile example.

warding. MV learns the frequency of meetings between nodes and visits to certain regions. The past frequencies are used to rank each bundle according to the likelihood of delivering a bundle through a specified path. MV determines a probability, $P_n^k(i)$, that the current node, k , can successfully deliver a bundle to a region i within n transfers. The probability is estimated by the following formula, assuming an infinite buffer at each node and N being the number of nodes in the network:

$$P_n^k(i) = 1 - \prod_{j=1}^N [1 - m_{j,k} P_{n-1}^j(i)]$$

where

$$P_0^k(i) = \frac{t_i^{(k)}}{t},$$

and $t_i^{(k)}$ is the number of rounds node k visited region i during the previous t rounds, and the meeting probability based on the meetings in the last t rounds is

$$m_{j,k} = \frac{t_{j,k}}{t}$$

and $t_{j,k}$ is the number of meets between nodes j and k in the same region. MV does not blindly forward to neighbors, but only forwards these bundles upon request from next-hop neighbors. Simulation results show that the packet delivery rate can be 50 percent higher under MV than a FIFO scheme.

The shortest expected path routing (SEPR) protocol is proposed in [30]. SEPR first estimates the link forwarding probability based on history data using the following formula:

$$P_{i,j} = \frac{Time_{connection}}{Time_{window}},$$

where $Time_{connection}$ is the time period the nodes i and j are connected and $Time_{window}$ is the sampling time window length. The shortest expected path is calculated based on the estimated link probability. When two nodes meet, the following routing protocol is performed. Each message stored in the cache is assigned an *effective path length*, EPL. The value is set at infinite when the message is first inserted in the cache. When the message is propagated to another node B, EPL is updated if the expected path length from B to the destination is smaller. A smaller value of EPL indicates a higher probability of delivery. Therefore, during the cache replacement process, those messages with smaller EPLs are removed first. EPL is also used in deciding which nodes to forward the messages. Their protocols were evaluated through simulation assuming certain types of mobility models. The probability estimation

does not rely on any location information. Under the algorithm, the same message could be forwarded to multiple nodes to increase reliability and reduce delay. Numerical results indicate that under SEPR, a 35 percent improvement of deliver rate and 50 percent reduction in resource cost can be achieved compared with epidemic [20] and DLE routing [26]. The gain comes from the fact that SEPR considers end-to-end performance while DLE only considers one-hop performance.

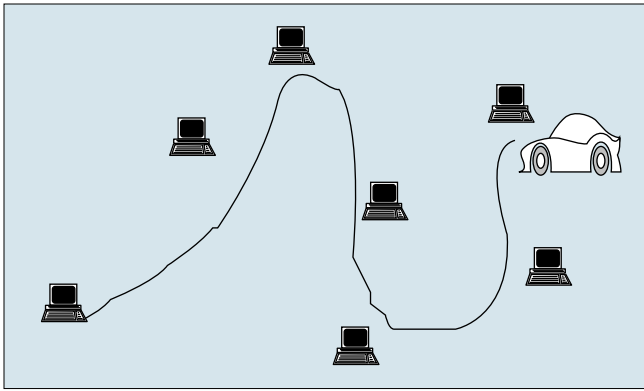
Similar to SEPR, minimal estimated expected delay (MEED) routing is proposed in [31]. It is an extension of the work by Jain [16] under the *contact summary oracle* assumption discussed earlier. The later model is a time-independent model (minimal expected delay, MED). MEED computes the expected delay using the observed contact history, in which a node records the connection and disconnection time of each contact over a sliding history window. When local link-state information changes, updates must be propagated to all nodes in the network. Epidemic link-state protocol is used for link-state exchange. The routing table is re-computed each time a contact arrives and before a message is to be forwarded, resulting in per contact routing. The difference between MEED and MED is that under MEED a decision is made with the most recent information possible, while under MED a decision is made offline using average information and will not change over time. MEED may have higher overhead and may result in loops. Care must be taken to prevent loops. How to choose the window size is not discussed in the paper, and the window size is treated as a control parameter. Numerical results show that in buffer constrained networks, MEED performs much better than Epidemic [31].

Anycast routing is very useful for many applications such as resource discovery in DTNs. In [52] the authors define the anycast semantics for DTN based on a new model. A novel metric named EMDDA (Expected Multi-Destination Delay for Anycast) and a corresponding routing algorithm for anycast routing DTNs are presented. Extensive simulation results show that the proposed EMDDA routing scheme can effectively improve the efficiency of anycast routing in DTNs.

MODEL-BASED APPROACH

In previous work, in estimating the forwarding probability, it is assumed that mobile devices move *randomly* without any specific knowledge about the trajectories. In the real world, however, devices do move following *certain known patterns* such as walking along a street or driving down the highway. Once users describe their motion pattern, the intermediate nodes have a more accurate estimation of which nodes move toward the destination with higher probability.

Model Based Routing (MBR) [32] uses world models of the mobile nodes for a better selection of relaying nodes and the determination of a receiver location without flooding the network. World models contain location information (e.g. road maps or building charts) and user profiles indicating the motion pattern of users (Fig. 3 in [32]). The key idea of the approach is to take into account that mobile devices typically do not follow the random walk motion pattern but are carried by human beings. Once humans describe their motion pattern or some sort of monitoring deduces it, MBR can rely on this



■ **Figure 4.** Control of node movement.

information in the form of user profiles to choose a relay that moves toward the target with higher probability. With the information of the receiver location, each intermediate node can determine the next relaying node based on the user profile. Each node offers an interface that emits the probability that the user will move toward a given location. Hence the routing algorithm can choose less relays if a small number of relays have been found that will move near or to the location with high probability. However, only a sketch of the algorithm is described in [32] (no detail is given). Obtaining the user profile is an open research question. Their work relies on the known receiver location, which is provided by a central location service, an unrealistic assumption.

A model of nodes moving along on a highway is described in [33]. With ad hoc networks deployed on moving vehicles, network partitions due to limited radio range become inevitable when traffic density is low, such as at night, or when few vehicles carry a wireless device. A key question to ask is whether it is possible to deliver messages in spite of partitions, by taking advantage of the fact that predictable node movement creates opportunities to relay messages in a store-and-forward fashion. The authors in [33] test the hypothesis that the motion of vehicles on a highway can contribute to successful message delivery, provided that messages can be relayed and stored temporarily at moving nodes while waiting for opportunities to be forwarded further. Messages are propagated greedily each time step by hopping to the neighbor closest to the destination. Two kinds of transmission schemes are used, *pessimistic forwarding* and *optimistic forwarding*, which are distinguished by how long the messages are permitted to stay in intermediate nodes. In pessimistic forwarding, a message is dropped whenever no next hop exists for its destination. This is how forwarding works in most ad hoc network implementations. In optimistic forwarding, messages without next hops may remain on intermediate nodes for some time, hoping that physical movement of network nodes eventually creates a forwarding opportunity. Using vehicle movement traces from a traffic micro simulator, the authors measure average message delivery time and find that it is shorter than when the messages are not relayed.

NODE MOVEMENT CONTROL-BASED APPROACHES

The approaches discussed in the previous sections let the mobile host wait passively for the network to reconnect. This may lead to unacceptable transmission delays for some applications. Some research works, therefore, have proposed approaches that try to limit these delays by exploiting and controlling node mobility. As illustrated in Fig. 4, the trajectories of some nodes (or special nodes) can be controlled so that overall system performance metrics, such as delay, can be improved.

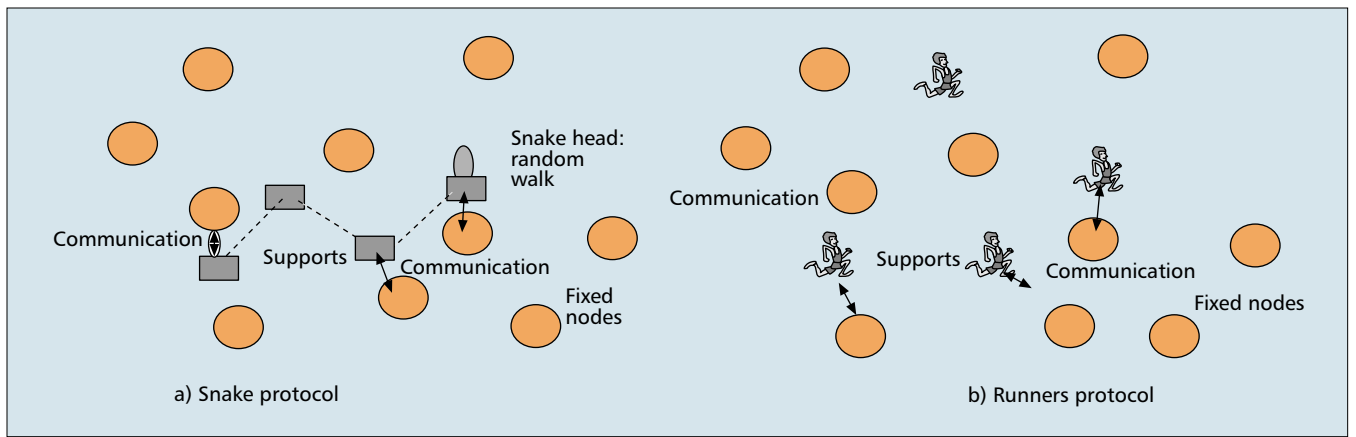
Li and Rus [34] explore the possibility of changing the host trajectories in order to facilitate communication in ad hoc networks. In contrast to letting the mobile host wait passively for reconnection, the mobile hosts actively modify their trajectories to minimize transmission delay of messages. Given an ad hoc network of mobile computers where the trajectory of each node is known, they develop an algorithm for computing a trajectory for sending a message from host A to host B by asking intermediate hosts to change their trajectories in order to complete a routing path between hosts A and B. The communication protocol proposed is an application-layer protocol (rather than a network-layer protocol). When the network cannot route a message to the destination due to a network partition, it will try to do an “up-call” for the scheme presented. Algorithms that minimize the trajectory modifications are developed under two different assumptions: the movements of all the nodes in the system are known, and the movements of the hosts in the system are not known. In the first case the problem is, given a mobile ad-hoc network, (which may be disconnected,) and the motion descriptions of the hosts (which is assumed to be known for all the movement of hosts), finding the shortest time strategy to send a message from one host to another. An optimal relay path algorithm is proposed, which computes a sequence of intermediate hosts that can relay the message to the destination. Intermediate nodes modify their trajectories in the smallest possible way. In the second case, they propose a method in which hosts *inform the other hosts* of their current positions. The key issues that need to be considered to make this approach work are

- When a host should send out information about its location update.
- To whom the host should send out this information.
- How the host should send out this information.

They model the communication problem in unknown mobile network environments by constructing a minimum spanning tree (MST), which contains the shortest edges in the graph that provide *full connectivity* in the graph. Each host has the responsibility of updating its location by informing all the hosts connected to it in the MST. However, their work assumes the network is almost fully connected; it is not quite clear what happens if no such MST is found.

Virtual mobile nodes (VMN), proposed in [53], is a distributed algorithm that runs on *abstract nodes that move in a predetermined and predictable manner*. In its simplest version, the virtual mobile nodes (VMN) travel through the network, collecting and delivering messages. In order to send a message, a real node examines its current location and calculates the current location of the VMN that is carrying out the service. *The node then waits until the virtual node is nearby and transmits the message to the virtual node*. The virtual node collects messages that it has received. Nodes that want to receive messages perform a similar protocol: if some node discovers that a VMN is nearby, it starts communication with the VMN. The real node’s movement is not controlled for the purpose of reducing message delay time.

In wireless networks, there usually are mismatches between available capacity and demand. When such a mismatch occurs in such networks, one way to add capacity is to increase the number of participants carrying bundles in the network. To achieve this, the work in [35] suggests the addition of a limited number of autonomous agents to the network area and studies the problem of augmenting the capacity of a DTN through autonomous agents that move in the network with the purpose of increasing network performance. The addition of these agents requires a control algorithm that can coordinate agent movements in order to optimize the performance of the network according to quality of service metrics desired by the



■ **Figure 5.** Illustration of the snake and runners protocols.

network administrator. The authors present a control-based approach and develop multi-objective controllers to control the mobility of autonomous agents. The design of control strategies *assumes the use of autonomous agents that can move to arbitrary locations in the physical environment*. Four controllers, *latency, bundle latency, unique bandwidth, and bandwidth*, are defined in [35]. Two approaches to multi-objective control, *subsumption* and *nullspace*, have been implemented and explored. Both techniques are from robotic research; nullspace controllers use linear algebra to coordinate controllers. Nullspace is defined as the set of inputs to a function where the value of the function does not change. *Nullspace composition* is used to coordinate collections of controllers. The controllers are ordered in a way such that a subordinate controller is forced to operate in the nullspace of controllers above it according to the order. The thresholded nullspace approach extends the nullspace approach to handle the networking situation that needed thresholded control. The subsumption approach differs from the nullspace approach in how the controllers dominate one another. Experimental results show that the *thresholded nullspace* approach outperforms the subsumption approach when resources are limited.

In [36] the authors describe a Message Ferrying (MF) approach for data delivery in sparse networks. MF is a proactive mobility assisted approach that utilizes a set of special mobile nodes called *message ferries* to provide communication services for nodes in the network. Similar to their real-life analog, message ferries move around the deployment area and take responsibility for carrying data between nodes. The main idea behind the Message Ferrying approach is to introduce non-randomness in the movement of nodes and exploit such non-randomness to help deliver data. Two variations of the MF schemes were developed, depending on whether ferries or nodes initiate non-random proactive movement. In the Node-Initiated MF (NIMF) scheme, ferries move around the deployed area according to known specific routes and communicate with other nodes they meet. With knowledge of ferry routes, nodes periodically move close to a ferry and communicate with that ferry. In NIMF, the ferry route is known by nodes, e.g., periodically broadcast by the ferry or conveyed by other out-of-band means. Nodes take proactive movement periodically to meet up with the ferry. As the sending node approaches the ferry, it forwards its messages to the ferry that will be responsible for delivery. The trajectory control mechanism of the node determines when it should proactively move to meet the ferry for sending or receiving messages. The difference between NIMF and VMN is that the nodes' movements are not controlled in VMN, while they are controlled in NIMF. In the Ferry-Initiated MF (FIMF) scheme, ferries move proactively to meet nodes. When a node wants to send packets to other nodes or receive packets, it generates a ser-

vice request and transmits it to a chosen ferry using a *long-range radio*. Upon reception of a service request, the ferry will adjust its trajectory to meet up with the node and exchange packets using short-range radios. In both schemes, nodes can communicate with distant nodes that are out of range by using ferries as relays. It is assumed that the ferry moves faster than nodes. In addition, it is assumed that nodes are equipped with a long-range radio that is used for transmitting control messages. Note that while the ferry is able to broadcast data to all nodes in the area, the transmission range of nodes' long-range radios may not necessarily cover the whole deployment area due to power constraints.

In [37] the authors study the problem of using multiple ferries to deliver data in networks with stationary nodes and designing ferry routes so that average message delay can be minimized. Multiple ferries offer the advantages of increasing system throughput (reducing message delay) and robustness to ferry failures. On the other hand, the route design problem with multiple ferries is more complicated than the single ferry case considering the possibility of interaction between ferries. The authors present ferry route algorithms for single ferry and multiple ferries cases, respectively. In the single ferry case, solutions for the well-studied traveling salesman problem are adopted. In the multiple ferries case, algorithms to assign nodes to specific ferries, synchronize among ferries, and assign ferries to specific routes are discussed. Simulation results are obtained to evaluate the performance of route assignment algorithms, especially on the effect of the number of ferries on the average message delay. Numerical results indicate that when the traffic load is low, the improvement in delay due to the increased number of ferries is modest. This is because the delay is dominated by the distance between nodes. However, when the traffic load is high, an increase in the number of ferries can significantly reduce the delay.

While the MV and FIMF approaches control the individual virtual nodes' trajectories, Chatzigiannakis *et al.* [38] present a snake protocol, where a snake-like sequence of carriers (called supports in [38], see Fig. 5a) or virtual nodes always remain pair-wise adjacent and move in a way determined by the snake's head. The head moves by executing a random walk over the area covered by the network. The nodes of supports move fast enough in a coordinated way so that they sweep (in sufficiently short time) the entire motion graph. Their motion and communication are accomplished in a distributed way via a support management sub-protocol. The supports play a moving backbone sub-network through which all communication is routed. The protocol consists of three components: *support motion subprotocol* P_1 ; *sensor subprotocol* P_2 ; and *synchronization subprotocol* P_3 . *Subprotocol* P_1 controls the motion of the supports in a distributed way. *Sensor subprotocol* P_2 notifies a sender that it may send its messages. *Syn-*

chronization subprotocol P_3 synchronizes all the nodes in the support. Results derived from an implementation show that only a small number of carriers is required for efficient communication.

Chatzigiannakis *et al.* [39] extend the work in [38] by presenting a new protocol, called the *runners* (Fig. 5b), where *each carrier* performs a random walk sweeping the whole area covered by the network, which is the only difference between the two protocols. The authors perform an experimental evaluation and comparison between the snake protocol and the runners protocol. It turns out that the runners protocol is more efficient (smaller message delays and memory requirements) and robust than the snake protocol. The authors also note that while the snake protocol is resilient only to one carrier failure, the runner protocol is resilient to up to M failures, where M is the number of carriers.

In DataMules [12] a three-tier architecture is proposed that connects spare sensors at the cost of high latency. At the top tier, there are access points or repositories that can be set at convenient places. The middle tier consists of DataMules that are mobile nodes (whose mobility pattern is not known) and can communicate with sensors and access points. DataMules have large storage capacity and renewable power. As DataMules move, they collect data from sensors and forward this data to the access points. The bottom tier consists of sensors that are randomly distributed across a region. To save energy, work performed by the sensors is minimal. DataMules are assumed to be capable of short-range wireless communication and can exchange data from a nearby sensor access point they encounter as a result of their motion (the movements of the DataMules are not predictable). Thus, DataMules can pick up data from sensors when in close range, buffer it, and drop off the data to wired access points when in proximity. The main advantage of the three-tier approach is the potential of large power savings by sensors because communication now takes place over a short range. Simple analytical models are presented to study the scaling of system characteristics as the system parameters, number of sensors, or number of DataMules change. Numerical results provide some relationship between the buffer requirements at the sensors (and at the DataMules) and the number of sensors (and the number of DataMules), respectively. It is observed that the change in the buffer capacity on each sensor should be greater than the number of DataMules so that the same success rate can be maintained.

CODING BASED APPROACHES

To cope with wireless channel loss, erasure coding and network coding techniques have recently been proposed for wireless ad hoc networks and DTNs. The basic idea of erasure coding is to encode an original message into a large number of coding blocks. Suppose the original message contains k blocks. Using erasure coding, the message is encoded into n ($n > k$) blocks such that if k or more of the n blocks are received, the original message can be successfully decoded. Here, $r = n/k$ is called the replication factor and determines the level of redundancy. Network coding comes from information theory and can be applied in routing to further improve system throughput. Network coding will be described in more detail later.

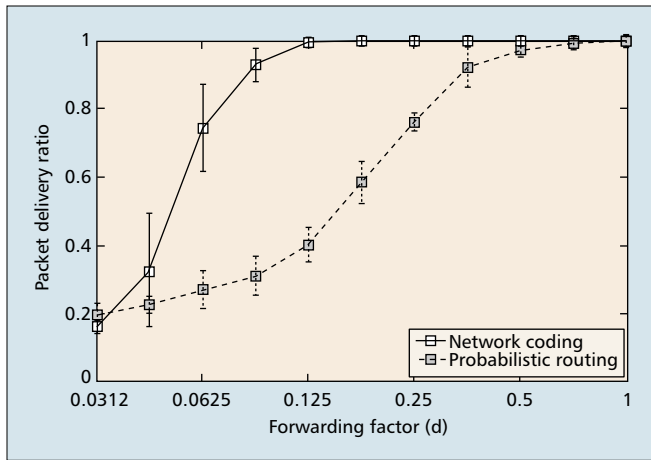
To extend the work in [16], the authors of [40] assume the probability, P_i , that the transmission over link i is successful (independent of other transmissions) is known. Given that the replication factor is r , they study the following allocation problem: to determine an optimal fraction, x_i , of the erasure code blocks that should be sent over path i , such that the probabili-

ty of successful reception is maximized. By fixing the replication factor and treating delay as a constraint, they formulate the problem as an optimal allocation problem and consider two cases of path failure scenarios: Bernoulli (0-1) path failure and partial path failure. In the Bernoulli path failure case, when a path fails, all the messages sent over the path are lost. In the partial path failure case, some messages can be recovered with certain probabilities. They prove that the optimal allocation problem under the Bernoulli path failure case, which is formulated as a mixed integer programming, is NP hard. For the partial path failure case, they first show that maximizing the successful probability is equivalent to maximizing the Sharpe-Ratio, which plays an important role in the theory of allocation assets in investment portfolios (see references in [40] for more details). They propose to use approximation approaches from economic theory to maximize the ratio. The solution of the optimal allocation problem is static and does not change over time, as it is assumed that the underlying path failure probability does not change over time.

Instead of optimally allocating a fixed portion of the coded blocks on each path from source to destination, it is proposed in [41] that coded blocks with replication factor r are equally split among the *first mr relays* (or contacts), for some constant m , and those relays must deliver the coded blocks to the destination *directly*. The original message can be decoded as soon as m contacts deliver their data (that is, as soon as $1/r$ of the coded blocks have been received). The difference between this approach and the one presented in [40] is that this approach sends data dynamically to the first m contacts the node meets. (In other words, the allocation of the coded blocks is not fixed.) It also differs from the estimation-based approaches discussed earlier in that it does not attempt to find which contacts have better chances to deliver the data. Instead, it simply forwards to the first m contacts the node meet (all contacts are equally good relays). Both analytic and simulation results show the erasure coding-based forwarding in DTNs significantly improves the worst case delay (compared with several other simple forwarding schemes).

To further improve the performance of the forwarding protocols-based erasure coding, in [43] the authors propose to combine erasure coding and estimation-based forwarding, which is referred to as estimation-based erasure coding (EBEC). The original messages are first encoded (using an erasure coding scheme). The encoded messages are forwarded to different relays that have higher a chance of delivering the messages. Numerical results show that EBEC outperforms the scheme studied in [41].

A probabilistic forwarding approach based on network coding is proposed for DTNs in [42]. Recently network coding has drawn much attention in the networking research community. Instead of simply forwarding packets received, intermediate nodes can combine some of the packets received so far and send them out as a new packet. For example, suppose that there are three nodes, A, B, and C. Nodes A and C want to exchange information through the middle node B. Node A first transmits packet x to node B, and node C transmits packet y to node B. Node B broadcasts x XOR y (not x and y in sequence). Since node A has packet x , and node C has packet y , node A can decode y and node C can decode packet x . For this example, it is easy to see that the number of transmissions is reduced when network coding is used. The basic idea in [42] is to use network coding to generate new packets. A coding vector is attached to each new packet. When a packet is received at a node, d new packets are generated and broadcast to the neighbors of the node, where d is referred to as a forwarding factor. When enough packets are received at the receiver, the original packet can be decoded. The value of d



■ **Figure 6.** Packet deliver ratio versus forwarding factor.

depends on the node density. Simulation results show that, for the given network setting in [42], for example, the packet deliver ratio using network coding is much higher than that under probabilistic forwarding, and most of the packets are delivered with lower forwarding factor (see Fig. 6 in [42]).

DISCUSSIONS AND FUTURE RESEARCH ISSUES

In the previous sections, we classify routing protocols into different categories: deterministic routing, random forwarding, history-based forwarding, etc. We refer to protocols in which the movement of nodes is not controlled as reactive protocols and protocols in which movements of the nodes can be controlled as proactive. Reactive and proactive protocols are summarized in Tables 1 and 2, respectively. In Table 1 the reactive protocols are compared based on the following metrics: buffer requirement, method of link state estimation, complexity (neighbor information exchange, computation for the link state forwarding, and location service required). In Table 2 proactive protocols are compared in terms of whether the movements of the nodes are controlled, the movements of special nodes are controlled, and information needed for the protocols.

Based on the existing work and the unique characteristics of the DTN, it is apparent that many research issues remain to be solved in the area of DTNs. In this section, we list some of the research issues that should be addressed and hope this will stimulate activity in the research community.

1. What is the proper objective function in designing a protocol in DTNs: short delay or high throughput, or others? Related questions are how to define the system capacity in such an intermittently connected network.

2. Methods to determine how many nodes to forward should be developed. There is a tradeoff. The larger the number of nodes forwarded to, the better the chance for packets to reach their destination, but the more network resources (bandwidth and buffer space) are needed. Analytical models should be developed, if possible, and simulation results should be obtained to quantify the tradeoff.

3. When multiple copies of the packets are in the network, duplication of packets occurs and such duplication requires a method of eliminating unnecessary copies to reduce the buffer occupancy. Where should the duplication reduction be done, at the destination or intermediate nodes and how? When original packets are received successfully at the receiver, how should intermediate nodes be informed to discard these packets. Informing intermediate nodes requires extra resources. Again, there is a tradeoff between efficiency and additional overhead.

4. Scheduling becomes much more complex in DTNs than in IP-centric networks, because connections in DTNs are intermittent while in normal IP networks they are not. Appropriate buffer management schemes (which packets to discard when full) and scheduling should be developed. One possible approach is to have separate queues for different outgoing links. Those packets whose destination will be disconnected soon (if known) should be scheduled to transmit first.

5. Whenever possible, information about node location and future movement should be utilized in designing the protocols. The forwarding protocols should leverage simple and accurate link availability estimation methods to make intelligent decisions, if feasible. There are some papers dealing with estimation of link availability for ad hoc networks [49, 54]. How to define user profiles and how to use them to estimate the deliver probability is also an open issue.

6. New security mechanisms must be developed, as techniques that rely on access to a centralized service cannot be used, or the assumption that all intermediate nodes are trusted is not valid.

7. Self-learning and automation algorithms should be developed so the underlying network is cognitive, and thus intelligent decisions on scheduling and forwarding can be made automatically.

8. Open spectrum [55] allows secondary users to opportunistically explore unused licensed band on a non-interfering basis. New algorithms to utilize those unused channels (resulting in intermittent connectivity) dynamically and efficiently should be developed.

9. Transmissions in networks with directional antennas are often pre-scheduled and may result in intermittent connectivity [56]. Power management in energy-aware network (range and/or wake/sleep periods control) may also result in intermittent connectivity. Therefore, scheduling transmissions with directional antennas and power management should take into account the DTN requirements or characteristics.

10. As the mobility of nodes in a mobile ad hoc network might lead to network partitions, and directional antennas can transmit over longer distances, in [57] the authors propose to use directional antennas to bridge such partitions when needed. The basic idea behind this method is to use the capability of a directional antenna to transmit over longer distances, but to adaptively use this capability only when necessary for selected packets. Methods to close broken links in ad hoc networks should be developed to cope with partial connectivity.

CONCLUSIONS

In this article we provide an overview of the state of the art on routing protocols in DTNs. Many excellent approaches to addressing the unique problems in DTNs have been reported in the literature. Each approach has its own advantages and disadvantages in terms of network efficiency and resources required. Even though the network behavior may not be known, the performance of these protocols developed under the deterministic case may serve as bounds on the network performance and as guidelines in designing protocols for the stochastic case. It seems that dynamic forwarding protocols based on the latest contact information combining with history information may perform better in the stochastic case. The main contributions of this article are

- Classifying different routing protocols into different categories according to the characteristics of the network behaviors and the corresponding protocols proposed.
- Identifying many open research issues in the routing area in DTNs.

Protocol	Buffer management (if buffer size is finite)	Estimation of link forwarding prob.	Use location or future information	Remarks	Complexity	Year of pub.
Epidemic [20]	Infinite	No	No	First work in DTN routing	Simple	2000
SWIM [22]	Infinite	No	No	Similar with [20], the probability to forward on a link can be controlled	Simple	2003
MRP [23]	Remove least recent (forwarding it randomly to one)	No	No	Broadcast once, then randomly forward to one node if buffer is full	Simple; integrate with existing routing protocols	2003
Spraying [24]	Infinite	No	Location	Forwarding once, then spraying	Simple; central location tracking,	2001
DLE [26]	DOA, DRA, DLE, DLR	Yes	No	Forwarding packets only upon <i>request</i> from next hop nodes	Exchange and computation	2001
PROPHET [27]	Infinite	Yes		Require two neighboring nodes exchanging routing information,	Exchange and computation	2003
MV [35]	Yes, delete those packets other nodes having higher probability	Yes, calculate the probability of delivering the bundle in h hops to destination		Need to estimate the likelihood two nodes meet	Exchange and computation	2005
SEPR [30]	Yes, remove those packets with smaller EPL	Yes, through measurement	No		Computation only	2003
CAR [28]	Infinite	Yes, Kilman filter	No	Integrate with DSDV	Computation only	2005
MBR [32]	Infinite	User profile assumed given	Yes, both	No details, only sketch of the algorithm presented	Exchange and computation; central location service	2001
IBRR [29]	Infinite	Yes, require two nodes exchange	Yes, both	Requires two-hop information exchange	Exchange and computation; location tracking, velocity, mobility	2002
Erasure coding [41]	Infinite	No	No	Source nodes enlarge the packets to be sent	Requires extra computation for decoding	2005

■ Table 1. Summary and comparison of the reactive approaches.

This will assist readers who are new to the field to have a better overall understanding of the state of the art in this area and assist them to start conducting research quickly. Two other related research areas (not covered here) are estimation methodologies for link availability and partition prediction (see [49, 54, 58]) and how to provide efficient data dissemination in partially connected networks (see [59]).

DTN research, in general, is still in its early stages and there are still many open issues that need to be resolved before the benefits of the DTNs can be fully utilized. Wireless networks with high mobility and short radio range will become a common phenomenon, and therefore it is imperative that those

issues be fully understood and studied. As it is still in its early stages of research, the purpose of this article is to summarize the current status of an evolving research community. It is the author's desire that this summary can provide a jump-start for other new researchers in this area and motivate the research community in developing new efficient and better protocols.

ACKNOWLEDGMENTS

The author would like to sincerely thank the anonymous reviewers for their time reading the article and for their valuable comments that improved the quality of the article signifi-

Protocol	Control nodes trajectory	Node movement patterns known	Special nodes' movement if needed	Remarks	Year of Pub.
MV [35]	No	No	Controlled	Extra nodes needed	2005
Li&Rus-1 [34]	Yes, intermediate nodes change their trajectories	Completely known	Not needed	The network is assumed to be almost connected; MST is not a realistic assumption in Li&Rus-2	2003
Li&Rus-2 [34]	Yes, based on information learned from updates	No	Not needed; MST used for updates		2003
NIFM [36]	Yes, node trajectory controlled (move closer to MF)	No	Predetermined		2004
FIFM [36]	No	No	Based on user's request (via long-range radio)	Long range radio is not a realistic assumption?	2004
VMN [53]	No (nodes wait to get closer to VMN)	No	Predetermined		2004
Snake [38]	No	No	Special nodes form a snake		2001
Runners[39]	No	No	Random walk		2001

■ Table 2. Summary and comparison of the proactive approaches.

cantly. Also thanks Dr. Dell Kronewitter of SDRC for providing valuable comments and proofreading the manuscript.

REFERENCES

- [1] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)," RFC 3626, IETF Network Working Group, October 2003.
- [2] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," RFC 3561, IETF Network Working Group July 2003.
- [3] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," Imielinski and Korth, Eds., *Mobile Computing*, vol. 353, Kluwer Academic Publishers, 1996.
- [4] Y.-B. Ko and N. Vaidy, "Location-Aided Routing in Mobile Ad Hoc Networks," *ACM Wireless Networks Journal*, June 2000.
- [5] M. Grossglauser and M. Vetterli, "Locating Nodes with EASE: Mobility Diffusion of Last Encounters in Ad Hoc Networks," *INFOCOM*, 2003.
- [6] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli, "Age Matters: Efficient Route Discovery in Mobile Ad Hoc Networks using Encounter Age," *MobiHoc*, 2003.
- [7] S.-J. Lee, W. Su, and M. Gerla, "Wireless Ad Hoc Multicast Routing with Mobility Prediction," *Mobile Networks and Applications J.*, 6/2001.
- [8] Y. Ge, Thomas Kunz, and Louise Lamont, "Quality of Service Routing in Ad Hoc Networks Using OLSR," *HICSS 2003*.
- [9] J. J. Garcia-Luna-Aceves, M. Mosko, and C. Perkins, "A New Approach to On-Demand Loop-Free Routing in Ad Hoc Networks," *Proc. 22nd ACM Symp. Principles of Distributed Computing (PODC 2003)*, Boston, Massachusetts, July 13–16, 2003.
- [10] S. Burleigh et al., "Delay-Tolerant Networking: An Approach to Interplanetary Internet," *IEEE Commun. Mag.*, June 2003.
- [11] P. Juang et al., "Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet," *Proc. ASPLOS*, Oct. 2002.
- [12] R. Shah et al., "Data MULEs: Modeling a Three-tier Architecture for Sparse Sensor Networks," *IEEE SNPA Wksp.*, May 2003.
- [13] A. Pentland, R. Fletcher, and A. A. Hasson, "A Road to Universal Broadband Connectivity," *2nd Int'l Conf. Open Collaborative Design for Sustainable Innovation*; Development by Design, Dec. 2002.
- [14] Wizzy Project, <http://www.wizzy.org.za/>
- [15] DTN Research Group, <http://www.dtnrg.org/docs/specs/draft-irtf-dtnrg-arch-04.txt>
- [16] S. Jain et al., "Routing in Delay Tolerant Network," *ACM SIGCOM'04*, Portland, Oregon, 2004.
- [17] K. Fall, "A Delay-Tolerant Network Architecture for Challenged Internets," *Proc. SIGCOMM'03*, Aug. 2003.
- [18] S. Merugu et al., "Routing in Space and Time in Networks with Predicable Mobility," Georgia Institute of Technology, Technical Report, GIT-CC-04-7, 2004.
- [19] R. Handorean et al., "Accommodating Transient Connectivity in Ad Hoc and Mobile Settings," *Pervasive 2004*, Apr. 21–23, 2004, Vienna, Austria, pp. 305–22.
- [20] A. Vahdat and D. Becker, "Epidemic Routing for Partially Connected Ad Hoc Networks," Tech. Rep. CS-200006, Department of Computer Science, Duke University, Durham, NC, 2000.
- [21] M. Grossglauser and D. Tse, "Mobility Increases the Capacity of Ad Hoc Wireless Networks," *INFOCOM*, 2001.
- [22] T. Small and Z. J. Haas, "The Shared Wireless Infostation Model — A New Ad Hoc Networking Paradigm (or Where there is a Whale, there is a Way)," *Mobihoc 2003*, June 1–3, 2003.
- [23] D. Nain et al., "Integrated Routing and Storage for Messaging Applications in Mobile Ad Hoc Networks," *Proc. WiOpt*, Autiplus, France, Mar. 2003.
- [24] F. Tchakountio and R. Ramanathan, "Tracking Highly Mobile Endpoints," *ACM Wksp. Wireless Mobile Multimedia (WoW-MoM)*, July 2001, Rome, Italy.
- [25] J. Su et al., "User Mobility for Opportunistic Ad-Hoc Networking," *6th IEEE Wksp. Mobile Comp. Sys. and Applications (WMCSA)*, UK, Dec. 2004.
- [26] A. Davids, A. H. Fagg, and B. N. Levine, "Wearable Computers as Packet Transport Mechanisms in Highly-Partitioned Ad-Hoc Networks," *Proc. Int'l. Symp. Wearable Comp.*, Zurich, Oct. 2001.
- [27] A. Lindgren et al., "Probabilistic Routing in Intermittently Connected Networks," *Mobile Comp. and Commun. Rev.*, vol. 7, no. 3, July 2003.

- [28] M. Musolesi et al., "Adaptive Routing for Intermittently Connected Mobile Ad Hoc Networks," *IEEE WoWMoM 2005*.
- [29] C. Shen et al., "Interrogation-Based Relay Routing for Ad Hoc Satellite Networks," *IEEE Globecom '02*, 2002.
- [30] K. Tan, Q. Zhang, and W. Zhu, "Shortest Path Routing in Partially Connected Ad Hoc Networks," *IEEE Globecom 2003*.
- [31] Jones, Li, and Ward, "Practical Routing for Delay Tolerant Networks," *SIGCOMM05-DTN Wksp. '05*.
- [32] C. Becker and G. Schiele, "New Mechanisms for Routing in Ad Hoc Networks," *4th Plenary Cabernet Wksp.*, Pisa, Italy, Oct. 2001.
- [33] Z. Chen et al., "Ad Hoc Relay Wireless Networks over Moving Vehicles on Highways," *ACM Mobihoc 2001*.
- [34] Q. Li and D. Rus, "Communication in Disconnected Ad Hoc Networks Using Message Relay," *J. Parallel Distributed Computing*, 63, 2003, pp. 75–86.
- [35] B. Burns et al., "MV Routing and Capacity Building in Disruption Tolerant Networks," *IEEE INFOCOM 2005*, Miami, FL, Mar., 2005.
- [36] W. Zhao et al., "A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks," *Proc. 5th ACM Int'l. Symp. Mobile Ad Hoc Net. and Comp.*, ACM Press, 2004, pp. 187–98.
- [37] W. Zhao, M. Ammar, and E. Zegura, "Controlling the Mobility of Multiple Data Transport Ferries in a Delay-Tolerant Network," *INFOCOM*, 2005.
- [38] I. Chatzigiannakis et al., "Analysis and Experimental Evaluation of An Innovative and Efficient Routing Protocol for Ad-Hoc Mobile Networks," *Lecture Notes in Computer Science*, vol. 1982, 2001, pp. 99–111.
- [39] I. Chatzigiannakis et al., "An Experimental Study of Basic Communication Protocols in Ad-Hoc Mobile Networks," *Lecture Notes in Computer Science*, vol. 2141, 2001, pp. 159–69.
- [40] S. Jain et al., "Using Redundancy to Cope with Failures in a Delay Tolerant Network," *SIGCOMM'05*.
- [41] Y. Wang et al., "Erasure-Coding Based Routing for Opportunistic Networks," *SIGCOMM DTN Wksp. 2005*.
- [42] J. Widmer and J. Le Boudec, "Network Coding for Efficient Communication in Extreme Networks," *SIGCOMM DTN Wksp. 2005*.
- [43] Y. Liao et al., "Combining Erasure-coding and Relay Node Evaluation in Delay Tolerant Network Routing," Microsoft Technical Report, MR-TR-2006-05, also submitted to *IWCMC 2006*, Jan. 2006.
- [44] A. Ferreira, "Building a Reference Combinatorial Model for MANETs," *IEEE Network*, Sept./Oct. 2004.
- [45] L. Ford and D. Fulkerson, *Flows in Networks*, Princeton University Press, 1962.
- [46] A. Iacono and C. Rose "Infostations: New Perspectives on Wireless Data Networks," WINLAB, Technical Document, Rutgers University, 2000.
- [47] A. Spyropoulos et al., "Spary and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks," *ACM SIGCOMM Wksp. Delay Tolerant Networking (WDTN-05)*, Aug. 2005.
- [48] Akis Spyropoulos et al., "Single-copy Routing in Intermittently Connected Mobile Networks," *IEEE SECON*, Oct. 2004.
- [49] A. B. McDonald and T. Znati, "A Mobility-based Framework for Adaptive Clustering in Wireless Ad Hoc Network," *IEEE JSAC*, Aug. 1999.
- [50] C. Perkins, *Ad hoc Networking*, Addison Wesley, Reading, MA, 2001.
- [51] Y. Wang and H. Wu, "DFT-MSN: The Delay Fault Tolerant Mobile Sensor Network for Pervasive Information Gathering," *IEEE INFOCOM'06*, Apr. 2006.
- [52] Y. Gong et al., "Anycast Routing in Delay Tolerant Networks," Technical Report MSR-TR-2006-04, Microsoft Research, Jan. 2006 <ftp://ftp.research.microsoft.com/pub/tr/TR-2006-04.pdf>
- [53] S. Dolev et al., "Virtual Mobile Nodes for Mobile Ad Hoc Networks," *18th Int'l. Symp. Distributed Comp.*, DISC 2004.
- [54] S. Jiang et al., "A Prediction-based Link Availability Estimation for Mobile Ad Hoc Networks," *IEEE INFOCOM 2001*.
- [55] R. Berger, "Open Spectrum: A Path to Ubiquitous Connectivity," *ACM Queue*, May 2003.
- [56] Z. Zhang, "DTRA: Directional Transmission and Reception Algorithms in WLANs with Directional Antennas for QoS Support," *IEEE Networks*, May/June 2005.
- [57] A. Saha and D. Johnson, "Routing Improvement Using Directional Antennas in Mobile Ad Hoc Networks," *IEEE Globecom 2004*.
- [58] P. Samar and S. Wicker, "On the Behavior of Communication Links of a Node in a Multi-Hop Mobile Environment," *ACM MobiHoc 2004*.
- [59] G. Karumanchi et al., "Information Dissemination in Partitionable Mobile Ad Hoc Networks," *Proc. IEEE Symp. Reliable Distributed Systems*, Lausanne, Switzerland, Oct. 1999.

BIOGRAPHY

ZHENSHENG ZHANG (zzhang@ieee.org) received his Ph.D. in electrical engineering from UCLA. He is currently with the San Diego Research Center (SDRC), Principal Investigator for several DOD projects. Before joining SDRC he visited Microsoft Research in the summer of 2002 and has worked at Sorrento Networks, Bell Laboratories, Lucent Technologies, and Columbia University. He has published more than 100 papers in *ACM/IEEE Transactions on Networking*, *IEEE Journal on Selected Areas in Communications*, *IEEE Transactions on Communications*, and key ACM/IEEE conferences. Currently, Dr. Zhang is an editor of *IEEE Transaction on Wireless Communications*. He served as the General Chair of the Broadband Wireless Networking Symposium (October 2004). He also served as a Guest Editor for the *IEEE Journal on Selected Areas in Communications* special issue on overlay networks (2003) and the *Journal of Wireless Networks* issue on multimedia wireless networks (August 1996). His research interests include wireless ad hoc networks and wireless sensor networks.