10

Delay-Tolerant Networks in VANETs

Yifeng Shao,
Cong Liu, and Jie Wu
Florida Atlantic University

10.1 Introduction

In many commercial applications$^{1-4}$ and in road safety systems, vehicular delay-tolerant networks have been envisioned to be useful. For example, a vehicular ad hoc network (VANET) can be used to alert drivers of traffic jams ahead, help balance traffic loads, and reduce traveling time. It can also be used to propagate emergency warnings to drivers behind the vehicles in an accident in order to prevent compounding on accident that has already taken place. Transportation safety issues have been addressed in Refs. 1 and 3, where vehicles communicate with each other and with static network nodes such as traffic lights, bus shelters, and traffic cameras.

The Federal Communications Commission (FCC) has allocated 75 MHz of spectrum for short-range vehicle-to-vehicle or vehicle-to-roadside communications. IEEE is working on standard specifications for intervehicle communication. In the near future, intervehicle communication will be enabled by communication devices equipped in general vehicles and form a large-scale VANET.

The cost of a wireless infrastructure is high and may not be possible when such an infrastructure does not exist or is damaged. Although services can be supported by a wireless infrastructure, from the service provider point of view, setting up a wireless LAN is very cheap, but the cost of connecting it to the Internet or the wireless infrastructure is high. From the user point of view, the cost of accessing data through a wireless carrier is still high and most cellular phone users are limited to voice services. Moreover, in the event of a disaster, the wireless infrastructure may be damaged, whereas wireless LANs and vehicular networks can be used to provide important traffic, rescue, and evacuation information to the users.

Many researchers and industry players believe that the benefit of vehicular networks for traffic safety and many commercial applications[1–3] should be able to justify the cost, although the cost of setting up vehicular networks is high. In the near future, many of the proposed delay-tolerant data delivery applications can be supported with such a vehicular delay-tolerant network already in place.

The fact that vehicular networks are highly mobile and sometimes sparse complicates multihop delay-tolerant data delivery through VANETs. The network density is related to traffic density. Traffic density is affected by location and time. It is low in rural areas and at night time, but very high in largely populated areas and during rush hours.

Finding an end-to-end connection is very difficult for a sparsely connected network. Opportunities for mobile vehicles to connect with each other intermittently while moving is introduced by the high mobility of vehicular networks. There are ample opportunities for moving vehicles to set up a short path with few hops in a highway model, as shown by Namboodiri et al.[5] A moving vehicle can carry a packet and forward it to the next vehicle. The message can be delivered to the destination without an end-to-end connection for delay-tolerant applications through store-carry-and-forward.

This chapter studies the problem of efficient data delivery and dissemination in vehicular delay-tolerant networks.

10.2 Overview

The rest of this chapter is organized as follows. First, we review the most up-to-date research regarding delay-tolerant networks (DTNs), with a focus on the routing problem. After that, we illustrate the car-following vehicle traffic model, which appropriately represents the mobility pattern of VANETs and which has a significant impact on the performance of specific data dissemination algorithms. Based on the traffic model, the problem of data dissemination is studied. We categorize the data dissemination problem into two aspects: vehicle-to-roadside and vehicle-to-vehicle (V2V). In the V2V case,

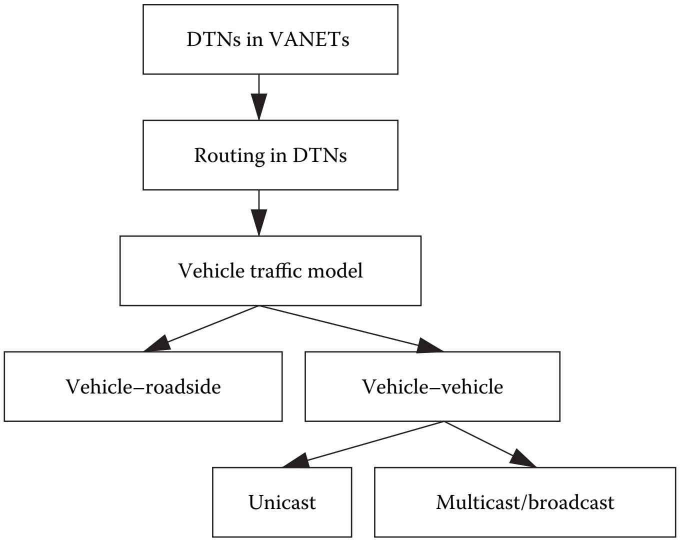

FIGURE 10.1 The structure of this chapter.

we study both the unicast problem and the multicast/broadcast problem. The structure of this chapter is organized as in Figure 10.1.

10.3 Delay-Tolerant Networks

As part of the Internet Research Task Force (IRTF), The Delay-Tolerant Networking Research Group (DTNRG)[6] was created to address the architectural and protocol design principles needed for interconnecting networks operating in environments where continuous end-to-end connectivity is sporadic.

DTNRG members are defining the initial DTN architecture. Kevin Fall was among the first to describe the main challenges facing current IP-based networks.[7] He proposed a DTN communication architecture based on a message-oriented overlay implemented above the transport layer. Messages are aggregated in "bundles" that form the protocol data units in a virtual message-switching architecture. Devices that implement this bundle layer, called DTN nodes, use persistent storage-to-buffer bundles whenever a proper contact is not available for forwarding.

Reliable delivery and optional end-to-end acknowledgment is implemented by the bundle layer. In addition, the bundle layer also implements security services and a flexible naming scheme with late binding. For more details on the DTN architecture, the reader should consult Ref. 7 and the Internet Draft by Vint Cerf et al.[8] Because the bundle layer is implemented above several transport layers, it supports interconnecting, heterogeneous networks using DTN gateways, similar to how Internet gateways route packets between networks with different data links.

Fall et al.[7] point out that routes in a DTN consist of a sequence of time-dependent communication opportunities, called contacts, during which messages are transferred from a source to the destination. Contacts are described by capacity, direction, the two endpoints, and temporal properties such as begin/end time and latency. Routing in a network with time-varying edges involves finding the optimal contact path in both space and time, meaning that the forwarding decision must schedule transmissions considering temporal link availability in addition to the sequence of hops to the destination.

This problem is exacerbated when contact duration and availability are nondeterministic. Contact types are classified in Refs. 7 and 8. Persistent contacts are always available. A scheduled contact is an agreement to establish a contact at a particular time for a particular duration. Opportunistic contacts present themselves unexpectedly. On-demand contacts require some action in order to instantiate, but then function as persistent contacts until terminated. Predicted contacts are based on a history of previously observed contacts or some other information.

Message forwarding requires scheduling in addition to next-hop selection because DTN routing must operate on a time-varying multigraph. To optimize the network performance, DTN routing must select the appropriate contact defined by a next-hop and a transmission time. If a contact is not known when a message is received from the upper layer, the bundle layer will buffer it until a proper contact occurs or until the message is dropped. In conditions of a DTN with sporadic contact opportunities, the main objective of routing is to maximize the probability of delivery at the destination while minimizing the end-to-end delay.

A sketch of the types of DTN routing protocols is illustrated in Figure 10.2. When the routing protocol has better information regarding the current state of the topology and its future evolution, the forwarding decision is more effective. At one end of the spectrum is deterministic DTN routing, where the current topology is known and future changes can be predicted. With deterministic routing, message forwarding can be

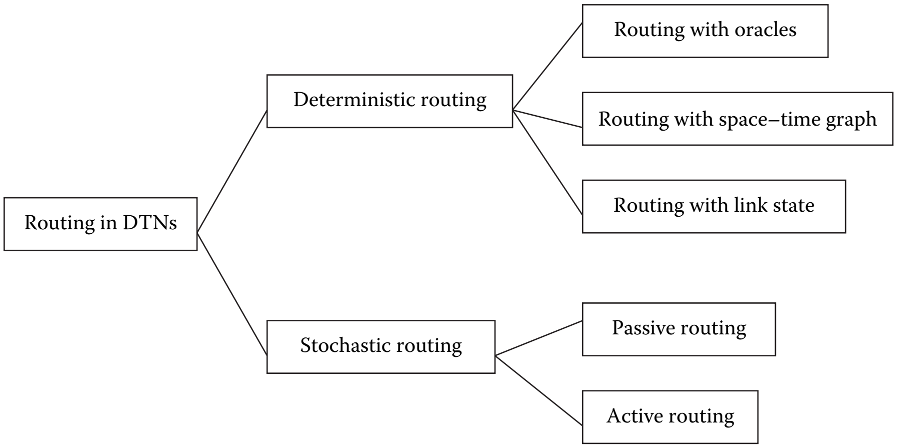

FIGURE 10.2 DTN routing protocols to be covered.

scheduled such that network performance is optimal and resource utilization is reduced by using unicast forwarding.

At the other end of the spectrum, node movement is random or unknown and nodes know very little or nothing about the future evolution of the topology. In this case, stochastic DTN routing forwards messages randomly hop by hop with the expectation of eventual delivery, but with no guarantees. In between, there are routing mechanisms that may predict contacts using prior network state information, or that adjust the trajectory of mobile nodes to serve as message ferries. Stochastic routing techniques rely more on replicating messages and controlled flooding for improving delivery rate, trading off resource utilization against improved routing performance in the absence of accurate current and future network states. The next section describes the principles of operation of representative deterministic and stochastic DTN routing mechanisms.[9]

10.4 Deterministic Delay-Tolerant Routing

In general, deterministic techniques are based on formulating models for time-dependent graphs and finding a space–time shortest path in DTNs by converting the routing problem to classic graph theory or by using optimization techniques for end-to-end delivery metrics. Deterministic routing techniques for networks with intermittent connectivity assume that local or global information on how the network topology evolves in time are available to a certain degree.

Good performance with less resource usage than stochastic routing techniques is provided by deterministic routing protocols using single-copy unicast for messages in transit. Deterministic routing mechanisms are appropriate only for scenarios where networks exhibit predictable topologies. This is true in applications where node trajectory is coordinated or can be predicted with accuracy, as in interplanetary networking.

10.4.1 Deterministic Delay-Tolerant Routing with Oracles

The distribution of network state and mobility information under sporadic connectivity, long delays, and sparse resources is a major problem facing deterministic routing protocols. In Ref. 10, Jain et al. present a deterministic routing framework that takes advantage of increasing levels of information on topology and traffic demand (oracles) when such information is predictable. A DTN multigraph is defined where vertices represent the DTN nodes and edges describe the time-varying link capacity between nodes. It is called a multigraph because multiple directed links between two nodes may exist.

One of the routing objectives is to minimize the end-to-end delay. Reducing the message transit times in the network also reduces contention for limited resources, such as buffer space and transmission time. Four knowledge oracles are defined: contacts summary oracle (for aggregate or summary contact statistics), contact oracle (for the time-varying contact multigraph), queuing oracle (for instantaneous queue state), and the traffic demand oracle (for present and future messages injected in the network). The authors adapt Dijkstra's shortest-path algorithm to support time-varying edge weights defined by the oracles available, and propose six algorithms for finding the optimal contact path.

Time-invariant edge weights is assumed in the first two algorithms in Ref. 10. The First Contact (FC) algorithm is a zero-knowledge approach that chooses a random edge to forward a message among the currently available contacts. If no contact is available, the message will be forwarded on the first edge that comes up. The Minimum Expected Delay (MED) algorithm applies the Dijkstra algorithm where the edge weight is time-invariant and is determined by the sum of the average waiting time (from the Contacts Summary oracle), propagation delay, and transmission delay. MED ignores congestion and does not recompute routes for messages in transit.

A time-varying edge cost, defined as the sum of the waiting, transmission, and propagation delays, is used in the following four proposed partial-knowledge algorithms. The waiting delay includes the time waiting for a contact and the queuing delay. The Earliest Delivery with Local Queuing algorithm (EDLQ) is equal to the local queue size at a particular node, and "0" for all other edges. EDLQ routes around congestion for the first hop and ignores queue occupancy at subsequent hops. Therefore, this algorithm must recompute the route at every hop. Cycles are avoided by using path vectors. Still, EDLQ is prone to message loss due to lack of available buffer space at reception.

The contacts oracle and the queuing oracle are used in the Earliest Delivery with All Queues (EDAQ) algorithm. EDAQ predicts the correct queue space for all edges at all times. In EDAQ, routes are not recomputed for messages in transit because the initial route accurately predicts all delays. EDAQ works only if capacity is reserved for each message along all contact edges. In practice, EDAQ is very difficult to implement in most DTNs with low connectivity, as it requires an accurate global distribution of queuing state. Limited connectivity also severely limits practical implementations of edge capacity reservations.

Simulation results indicate that algorithms that use the knowledge oracles (ED, EDLQ, and EDAQ) outperform the simpler MED and FC algorithms in terms of latency and delivery ratio. The more constrained the network resources are, the better the performance is for the algorithms that are more informed (i.e., use more oracles). A promising result is that routing with EDLQ (using only local queuing information) has a very similar performance to the EDAQ algorithm. This means that similar network performance can be achieved without expensive queue state dissemination and capacity reservations.

10.4.2 Deterministic Delay-Tolerant Routing with Space–Time Graphs

The trajectories and mission objectives of nodes may change. Therefore, in practice, contacts are deterministically predictable for only a finite time horizon. In Ref. 11, Merugu et al. propose a deterministic routing framework where a space–time graph is built from predicted contact information. It starts with a time-varying link function defined as "1" when the link between two nodes is available and "0" otherwise. This function is defined as a function of time, where the time is discretized.

In Ref. 11, the space–time graph is built in multiple layers where the network nodes are replicated at each layer for each time unit t. Each layer has a copy of each network node. A column of these vertices maps to a single network node. A temporal link in the space–time graph connects graph vertices from the same column at successive time intervals. When it is traversed, it indicates that the message is buffered. A spatial link connects two vertices from different columns, representing message forwarding. Forwarding delay is modeled by the number of layers traversed by a spatial link.

The objective of the least-cost routing in this DTN is to find the lowest cost (shortest) path from the source space–time node (column:layer) associated with the message arrival time to a vertex from the column corresponding to the destination DTN node. The end-to-end latency for a message becomes equal to the length of the path traversed in the space–time graph. The routing problem is solved using the Floyd–Wars Hall all-pairs shortest paths algorithm, modified to account for the particular characteristics of the space–time graph. Multiple message sizes are supported by a path-coloring scheme.

One issue with this approach is that time discretization increases the algorithm complexity by a factor of the size of the time horizon *T*. This space–time routing approach is similar to the Earliest Delivery partial knowledge algorithm from Jain et al.[10] in the way it handles queuing delays with route computation at each hop. Cycles are avoided by verifying the path vector from the message header when computing the next hop.

10.4.3 Delay-Tolerant Routing with Link State

In Ref. 12, Gnawali et al. propose ASCoT, a dynamic routing mechanism for space networks and the Positional Link State routing protocol (PLS) to implement position-based routing that enables the prediction of the trajectories of satellites and other space assets. Link state updates with predicted contacts and their link performances are disseminated in advance in the network through reliable flooding. Nodes execute a modified Dijkstra algorithm to recompute routing tables when link state updates are received.

In Ref. 13, the authors propose a data-centric approach similar to directed diffusion to support proximity routing for space assets in close formation. Note that in deterministic routing techniques using shortest-path algorithms, routing tables and forwarding schedules are recomputed whenever the contact graph state has changed, and selection of the next contact is done for a message at each hop along the path, as opposed to source routing. Thus, loops become possible because nodes may use outdated topology information. Cycles are avoided with path vectors.

For a limited range of applications, deterministic DTN routing protocols are effective where the contact schedule can be accurately modeled and predicted. Otherwise, it is necessary to frequently disseminate nodes' states throughout the network. In networks with constrained capacity or limited connectivity, this becomes very expensive and difficult to implement without an out-of-band broadcast channel. When contacts cannot be accurately predicted, routing must consider stochastic mechanisms that can only hint to predilection for future contacts based on historic information.

10.5 Stochastic Delay-Tolerant Routing

Depending on whether node mission is changed in order to support message relay, stochastic routing techniques can be passive or active. Passive routing techniques do not interfere with node missions and do not change node trajectory to adapt to traffic demands.

Passive routing techniques generally rely on flooding multiple copies of the same message with the objective of eventual delivery. In contrast, active routing techniques coordinate the mission (trajectory) of some nodes to improve capacity with their store-and-carry capability. In general, passive routing techniques trade off delivery performance against resource utilization.

By sending multiple copies of the same message on multiple contact paths, the delivery probability increases and the delay drops at the cost of additional buffer occupancy during message ferrying and higher link capacity usage during contacts. This approach is appropriate when very little or nothing is known about mobility patterns.

10.5.1 Passive Stochastic Routing

First, we present two passive stochastic routing protocols, Epidemic Routing and Spray and Wait, which do not need any information about the network state. For other routing protocols, nodes can memorize contact history and use it to make more informed forwarding decisions. The section then continues with several passive routing protocols that operate with contact estimation.

10.5.1.1 Epidemic Routing

In Ref. 14 Vahdat and Becker propose the Epidemic Routing protocol for message delivery in a mostly disconnected network with mobile nodes. Epidemic routing implements flooding in a DTN, named after a technique for message forwarding that emulates how a disease spreads through direct contact in a population during an epidemic. Even when just one individual of an entire population is initially infected, if the disease is highly contagious and contacts are frequent, over time it will spread exponentially and reach the entire population with a high probability.

In epidemic routing, the "disease" that spreads is a message that must reach one or more destinations. Each node maintains a summary vector with IDs of messages it has already received. When two nodes initiate a contact, they first exchange their summary vectors in the anti-entropy session. Comparing message IDs, each node decides what messages it has not already received that it needs to pull from other nodes.

The second phase of a contact consists of nodes exchanging messages. Messages have a time-to-live (TTL) field that limits the number of hops (contacts) they can pass through. Messages with $TTL = 1$ are forwarded only to the destination. The main issue with epidemic routing is that messages are flooded in the whole network to reach just one destination. This creates contentions for buffer space and transmission time.

Reserving a fraction of their storage for locally originated messages is an approach to mitigate buffer space contention for nodes. Even so, older messages in buffers will be dropped when new messages are received, reducing the delivery probability for destination nodes that have a low contact rate. An attempt to reduce resource waste is proposed that uses delivery confirmation (ACK) messages that are flooded starting from the destination and piggybacked with regular messages. Whenever a node receives an ACK, it purges the acknowledged message from its buffer, if it is still present.

Node movement is used in epidemic routing to spread messages during contacts. With large buffers, long contacts, or a low network load, epidemic routing is very effective and provides minimal delays and high success rates, as messages reach the destination on multiple paths. End-to-end delay depends heavily on nodes' contact rate (infection rate), which is in turn affected by the communication range and node speed.

To trade off message latency and delivery ratio, different implementations of epidemic routing tune message TTL and buffer allocation. In scenarios with a high message load, the increased contention from forwarding mostly redundant messages reduces the protocol performance.

Epidemic routing is relatively simple to implement and is used in the DTN research literature as a benchmark for performance evaluation.

10.5.1.2 Spray and Wait

In Ref. 15, Spyropoulos et al. present Spray and Wait, a zero-knowledge routing protocol introduced to reduce the wasteful flooding of redundant messages in a DTN. Similar to epidemic routing, this protocol forwards message copies to nodes met randomly during contact in a mobile network. The main difference from epidemic routing is that Spray and Wait limits the total number of disseminated copies of the same message to a constant number *L*. In the spray phase, for every message originated by a source, *L* copies are forwarded by the source and other nodes receiving the message upto a total of *L* distinct relays. In the wait phase, all *L* nodes storing a copy of the message perform direct transmission.

Direct transmission[16] is a single-copy routing technique in DTNs where the message is forwarded by the current node only, directly to the destination node. Direct transmission has been used for wildlife tracking applications and has minimal overhead, but suffers from unbounded delay as there is no guarantee that the source will ever have contact with the destination node.

Initially, Spray and Wait spreads *L* copies of a message in an epidemic fashion in order to increase the probability that at least one relay node would have direct contact with the destination node. With a simple Source Spray and Wait heuristic, the source node forwards all *L* copies to the first *L* nodes encountered.

Binary Spray and Wait is the optimal forwarding policy in which nodes move randomly with identical and independent probability distribution (i.i.d.). A message will be physically stored and transmitted just once even when a transfer may virtually involve multiple copies. Each message has a header field indicating the number of copies. The paths followed by copies of a message can be represented by a binary tree rooted in the source node.

The transfer contacts are formed by edges in the tree. The more nodes that have multiple copies to distribute, the less the expected end-to-end delay will be. The binary heuristic has the least expected delivery latency in networks with random i.i.d. random mobility. An interesting property of this routing protocol is that, as the network node count *M* increases, the minimum fraction *L*/*M* necessary to achieve the same performance relative to the optimal path decreases. This property makes the Spray and Wait approach very scalable.

10.5.1.3 PROPHET

Spray and Wait performs much better than epidemic routing at higher loads because the limit *L* of maximum transmissions reduces contention on queue space and transmission time. Some passive DTN routing protocols use delivery estimation to determine a metric for contacts relative to successful delivery, such as delivery probability or delay. Some of these protocols can forgo flooding and deliver single-copy messages by being selective with contact scheduling. The advantage is that considerably less memory, bandwidth, and energy are wasted on end-to-end message delivery.

One of the drawbacks of Spray and Wait is that nodes must keep track of other nodes' movements and contacts, and that network-wide dissemination of this information imposes additional overhead in a network that is already constrained. A representative routing protocol for DTNs that uses delivery estimation is PROPHET, a Probabilistic ROuting Protocol using History of Encounters and Transitivity, proposed by Lindgren et al. in Ref. 17. PROPHET works on the realistic premise that node mobility is not truly random. Instead, it is assumed that nodes in a DTN tend to visit some locations more often than others, and that node pairs that have had repeated contacts in the past are more likely to have contacts in the future.

A probabilistic metric called *delivery predictability* estimates the probability that node *A* will be able to deliver a message to node *B*. The delivery predictability vectors are maintained at each node *A* for every possible destination *B*.

Two nodes (*A* and *B*) exchange the summary vectors (as in epidemic routing) and also the delivery predictability vectors at the beginning of a contact. Node *A* then updates its own delivery predictability vector using the new information from *B*, after which it selects and transfers messages from *B* for which it has a higher delivery probability than *B*. The delivery probability is updated during a contact so that node pairs that meet more often have a higher value.

Additionally, the delivery predictability has a transitive property that encodes the **Q1** assumption that if nodes *A* and *B* have frequent contacts and nodes *B* and *C* have frequent contacts, then node *A* has a good chance of forwarding messages intended for node *C*. After exchanging delivery predictability vectors at the beginning of a contact, **Q2** nodes *A* and *B* update their values for each other node *C*.

As node *A* begins a contact with node *B*, it decides to forward a message to *B* with destination *C* if $P(B, C) > P(A, C)$. Node *A* will also keep a copy in its buffer. The buffer has a first-in first-out (FIFO) policy for dropping old messages when new messages are **Q3** received. Transitive reinforcement of delivery probabilities based on prior contacts make this protocol perform better in simulations than epidemic routing because it reduces the contention for buffer space and transmission time.

Related techniques for delivery probability estimation based on prior contact history are used in MV routing[18] and Zebranet.[19] A novel approach for delivery estimation is the use of a virtual Euclidean mobility pattern space, called MobySpace, proposed by Leguay et al.[20] The idea is that messages in a DTN should be forwarded to another node if this next hop has a mobility pattern similar to the destination node. This concept was adapted from the Content Addressable Network peer-to-peer overlay architecture.[21]

10.5.1.4 MobySpace

In existing works on user mobility in various scenarios where users tend to follow similar trajectories, the authors suggest a model where the node movement follows a power law. This means that the probability that a node is at a location *i* from a set of *N* locations is $P(i) = K(1/d)^{n_i}$, where n_i is the preference index for location i, $d > 1$ is the exponent of the power law, and K is a normalization constant. When d is high, nodes tend to visit far fewer locations far more often. When $d \to 1$, nodes have similar preference for all locations. The mobility pattern space has a dimension for each possible location, and the coordinate value of a node's point in this space (MobyPoint) in dimension i is equal to the probability $P(i)$. This model assumes that dwell time at each location is uniformly distributed in a narrow interval.

In MobySpace, two nodes that have a small distance between them are more likely to have a contact than two nodes that are situated further apart. With this insight, the forwarding algorithm simply decides to forward a message during a contact to a node that has a shorter distance to the message destination. Messages take paths through the MobySpace to bring them closer and closer to the destination. Several distance functions have been proposed to measure similarity in nodes' mobility patterns. The Euclidean and the cosine separation distance provide lower delays in simulations.

The MobySpace approach is only effective if nodes exhibit stable mobility patterns. It also fails if a message reaches a local maximum where the current node has a similar mobility pattern with the destination, but a direct contact with the destination is rare due to trajectory synchronization. Such a case is possible in a DTN where nodes are public transportation buses. Although the buses on a line follow the same path and visit the same stations, two buses may get within radio range only at night when they park in the garage.

Two nodes having similar mobility patterns does not mean that they are frequent contacts. A possible solution to this problem is to use the probability (or frequency) of direct contacts with the other nodes as dimensions in the MobySpace. Another approach to deal with the temporal variability of mobility patterns is to supplement MobySpace with conversion of the spatial visit patterns to the frequency domain, representing the dominant visitation frequency and the phase. Other issues with MobySpace include effective dissemination of location probabilities for all nodes in a constrained DTN and high convergence time.

10.5.2 Active Stochastic Routing

In active routing protocols, the trajectory of some nodes are controlled to improve delivery performance with store-and-carry. Mobile nodes pick up messages and ferry them for a distance before another contact brings them closer to the destination. Active routing techniques provide improved flexibility and lower delays with the additional cost of increased protocol and system complexity. Active DTN routing techniques are frequently implemented as optimization problems.

The general objective of an active routing protocol is to maximize network capacity, reduce message latency, and reduce message loss while facing resource constraints. Applications where mobile nodes are controlled to ferry messages can be used in multiple domains. In disaster recovery, mobile nodes (helicopters, UAVs, or personnel) equipped with communication devices capable of storing a large number of messages can be commanded to follow a trajectory that interconnects disconnected user partitions.

10.5.2.1 Meet and Visit (MV) Routing

In wireless sensor networks, mobile nodes can also traverse the sensing area and pick up/deliver measurements, queries, and event messages. In the remainder of this chapter, we review two DTN routing mechanisms that employ active node trajectory control. In Ref. 18, Burns et al. introduce the Meet and Visit (MV) routing scheme, where node trajectory is adjusted according to traffic demands by autonomous agents. MV aims to improve four performance metrics with a multi-objective control approach.

On each controlled mobile node, separate controllers for total bandwidth, unique bandwidth, delivery latency, and peer latency, respectively, are combined through multi-objective control techniques such as null-space or subsumption. Each controller adjusts the node trajectory such that its own objective is maximized.

The Total Bandwidth Controller selects the DTN that has the greatest number of unseen messages amortized by the trip time. This prevents making long trips without a matching load of new messages. The Unique Bandwidth Controller selects a node that has the largest number of new messages not yet forwarded to any other nodes. The Delivery Latency Controller picks the node with the highest average delivery time. The Peer Latency Controller selects the node that is least-visited by an agent s.t. the traveling time to visit this node does not increase the overall peer latency metric. **Q4**

The four controllers can be composed to optimize agent missions across performance metrics. To do that, controllers are first ordered according to their importance. With the null-space approach, an agent's subordinate controller actions can be optimized without affecting the performance of the dominant controller's actions. To increase the optimal solution space of the dominant controller, a minimum performance threshold method is used. The actions controlled by the subordinate controller are acceptable as long as the dominant controller's performance is above this threshold.

A different controller composition approach uses a subsumption approach. A controller with a higher priority computes the action space for achieving a specified performance level for its metric. Within this space, the immediate lower priority controller finds its own optimum without changing the performance of any higher priority controllers. MV implements an epidemic dissemination protocol for the network state necessary for the four controllers. Node information is tagged with a time stamp and flooded during contact.

Simulation results have shown that this approach is sufficient for low-bandwidth and latency-estimation errors, but not enough to correctly estimated "last visit" times and location information. MV routing could be further improved with additional offline or out-of-band network states. Another limitation of this approach is the key assumption that contact bandwidth is unlimited.

10.5.2.2 Message Ferrying

In Ref. 22, Zhao et al. describe a proactive Message Ferrying routing method (MF) with 2-hop forwarding and a single ferry. A message ferry is a special mobile node tasked with improving the transmission capacity in a mobile DTN. The authors present two methods for message ferrying in sparse DTNs. In the Node-Initiated Message Ferrying (NIMF) scheme the ferry follows a specific trajectory. Nodes that need to send messages adjust their trajectory periodically to meet the ferry for message up-/download.

The objective of the NIMF node trajectory control mechanism is to minimize message loss due to TTL expiration and buffer limits, while reducing the negative impact of trajectory changes on node mission goals. The first objective can be expressed by knowing message generation/drop rates and by estimating contact times. The second objective can be modeled as the Work Time Percentage (WTP). The WTP represents the fraction of time a node performs its main task. It is assumed that during a detour to meet a ferry, a node does not contribute to its main task. The NIMF controller allows node trajectory changes only when the WTP is above a minimum threshold.

In the Ferry-Initiated Message Ferrying (FIMF) scheme, the ferry responds to requests for contacts broadcast by nodes on a long-range radio channel. The authors show that the ferry trajectory control problem is NP-hard and propose a greedy nearest-neighbor heuristic and a traffic-aware heuristic that optimizes, locally, both location and message drop rates. In Ref. 23, the same authors extend their ferry-based DTN routing method for coordinating multiple message ferries such that traffic demands are met and delay is minimized. Approximations are provided for single-route and multi-route trajectory control. Ferry replacement algorithms for fault-tolerant delivery are further explored in Ref. 24.

10.6 Vehicle Traffic Model

In this section, we discuss vehicle traffic models. Vehicle traffic models are important for DTN routing in vehicle networks because the performance of DTN routing protocols are closely related to the mobility model of the network. The car-following model is used in civil engineering to describe traffic behavior on a single lane under both free-flow and congested traffic conditions.[25] This model assumes that each driver in the following vehicle maintains a safe distance from the leading vehicle and the deceleration factor is also taken into account for braking performance and drivers' behavior. The complete mathematical model is given by

$$S' = L + \beta' V + \gamma V^2$$

where S' is the headway spacing from rear bumper to rear bumper, L is the effective vehicle length in meters, and V is the vehicle speed in meters/second. β' is driver reaction time in seconds, and the γ coefficient is the reciprocal of twice the maximum average deceleration of a following vehicle. Both the β' parameter and the γ coefficient are introduced to ensure that the following vehicle can come to a complete stop if the leading vehicle suddenly brakes. As in many other civil engineering studies, we use a so-called "good driving" rule, which assumes that each vehicle has similar braking performance. In this case, the car following model can be simplified as

$$S' = L + \beta' V.$$

The car-following model has some limitations in modeling freeway traffic behavior for the purpose of wireless networking research, but is one of the most popular models in civil engineering. These limitations can be summarized as follows:

1. The car-following model is limited to the situation where driver reaction time is believed to be a dominant factor. Therefore, it is only an appropriate model under

free-flow traffic or heavy traffic scenarios. Empirical studies[26] confirm that during rush hour β' is typically a small number that represents the reaction time of a driver, following a log-normal distribution.[27] However, in light to moderate traffic, β' can be as large as 50 to 100 sec and cannot be interpreted as driver reaction time.[27] Instead, interarrival time between vehicles should be used to describe this spacing.

2. This is the focus of vehicular safety research in civil engineering. Therefore, the car-following model describes headway spacing between two adjacent vehicles of the same lane (i.e., lane-level spacing). From the network connectivity standpoint, however, we observe that the most relevant metric is spacing from the leading vehicle to the nearest following vehicle on a multilane road (i.e., road-level spacing), regardless of whether the following vehicle is on the same lane or on a different lane from the leading vehicle.

To address both of the aforementioned limitations, the car-following model is extended to the road level by replacing the lane-level reaction time β' with a road-level interarrival time β (the interarrival time of vehicles on any lane on the same road as observed from a fixed observation point). The lane-level car-following model can be generalized as

$$S = L_{\min} + \beta V$$

where $L_{\min}$ is the minimum spacing between any two adjacent vehicles, which is assumed to be zero in this study. By focusing on road-level intervehicle spacing S, the proposed model not only models rush-hour heavy traffic but also captures the sparse or intermediate traffic during nonrush hour times.

10.7 Vehicle–Roadside Data Access

Although a lot of research has been carried out on intervehicle communication, vehicle–roadside data access is also an important issue in vehicle DTN network. Medium access control (MAC) issues have been addressed in Refs. 2, 28, and 29, where slot-reservation MAC protocols[28,29] and congestion control policies for emergency warning[2] are studied.

In a recent paper on vehicle–roadside data access,[30] the roadside unit (RSU) can act as a router in a delay-tolerant network or as an access point for vehicles to access the Internet. Although this can bring many benefits to drivers, the deployment cost and maintenance cost are very high. As another option, RSU can also be used as a buffer point (or data island) between vehicles. This section focuses on the latter paradigm due to its low cost and easy deployment.

All data on the RSUs are uploaded or downloaded by vehicles in this paradigm. For example, some data, especially those with spacial/temporal constraints, only need to be stored and used locally. Applications that also belong to this case where the data is buffered at the RSUs and will not be sent to the Internet include the following:

1. *Real-time traffic.* Vehicles can observe real-time traffic observations and report them to nearby RSUs. The traffic data are stored at RSUs, providing real-time

query and notication services to other vehicles. The data can be used to provide traffic conditions and alerts such as road congestion and accidents.

2. *Value-added advertisement.* To provide efficient advertisements, stores may want to advertise their sale or activity information in nearby area. Without Internet connection,[4] they can ask the running vehicles to carry and upload the advertisement information to nearby RSUs. At the same time, other vehicles driving around can download these advertisements and visit the stores.
3. *Digital map downloading.* It is impossible for vehicles to install all the most up-to-date digital maps before traveling. This would help to solve the storage limitations of memory cards and changes resulting from frequent road construction. Hence, vehicles driving to a new area may update map data locally for travel guidance.

Vehicles are moving and they only stay in the RSU area for a short period of time. This makes vehicle networks different from traditional data access systems in which users can always wait for the service from the data server. As a result, there is always a time constraint associated with each request. Meanwhile, to make the best use of the RSU and to share the information with as many vehicles as possible, RSUs are often set at roadway intersections or areas with high traffic. In these areas, download (query) requests retrieve data from the RSU, and upload (update) requests upload data to the RSU. Both download and upload requests compete for the same limited bandwidth. As the number of users increases, deciding which request to serve at which time will be critical to system performance. Hence, it is important to design an efficient scheduling algorithm for vehicle–roadside data access.

10.7.1 A Model for Vehicle–Roadside Data Access

An architecture of vehicle–roadside service scheduling is shown in Figure 10.3, where a large number of vehicles retrieve (or upload) their data from (or to) the RSU when they are in communication range. The RSU (server) maintains a service cycle, which is non-preemptive; that is, a service cannot be interrupted until it finishes. When one vehicle enters the RSU area, it listens to the wireless channel.

All vehicles can send requests to the RSU if they want to access the data. Each request is characterized by a 4-tuple: <*v-id*, *d-id*, *op*, *deadline*>, where *v-id* is the identifier of the vehicle, *d-id* is the identifier of the requested data item, *op* is the operation that the vehicle wants to do (upload or download), and *deadline* is the critical time constraint of the request, beyond which the service becomes useless.

All requests are queued at the RSU server upon arrival. Based on the scheduling algorithm, the server serves one request and removes it from the request queue. Unlike traditional scheduling services, data access in vehicular networks has two unique features:

1. The arrival request is only active for a short period of time due to vehicle movement and coverage limitations of RSUs. When vehicles move out of the RSU area, the unserved requests have to be dropped.
2. Data items can be downloaded and uploaded from the RSU server. The download and update requests compete for the service bandwidth.

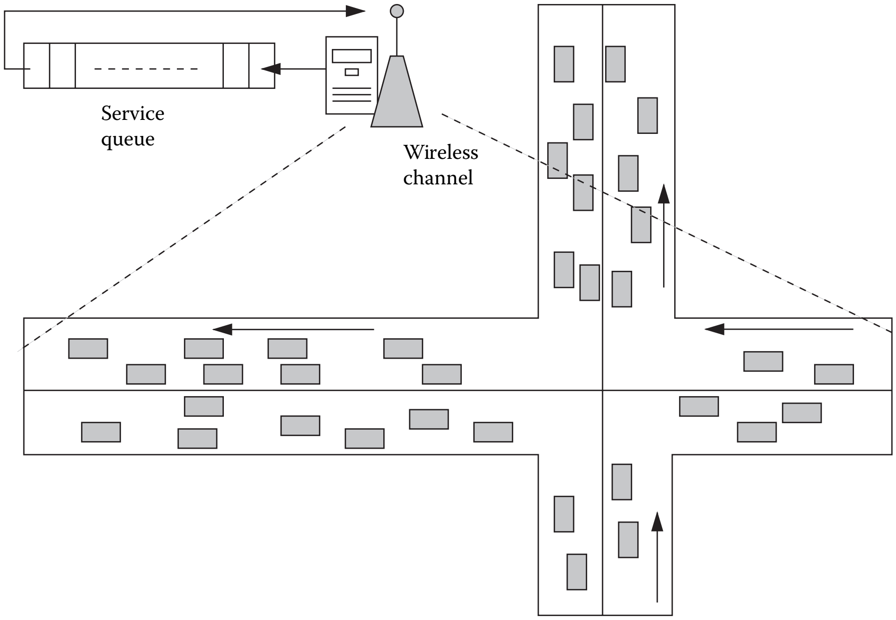

FIGURE 10.3 The architecture of vehicle–roadside service scheduling.

It is assumed that each vehicle knows the service deadline of its request. This is reasonable because when a vehicle with a GPS device enters the coverage area of a RSU, it can estimate its departure time based on the knowledge of its driving velocity and its geographic position. After a vehicle establishes connectivity with one RSU, it can get the geographic information and radio range of the RSU through beacon messages. With its own driving velocity and position information, the vehicle can estimate its departure time, which is its service deadline.

10.7.2 Performance Metrics

The metrics for scheduling algorithms are responsiveness (e.g., average/worst-case waiting time[31–33]) or fairness (e.g., stretch[34,35]) and are commonly used in previous works. In most of these works, requests do not have time constraints, and the data on the server is either not updated, or updated only by the server. However, in the vehicle–roadside data access scenario, requests that are not served within a set time limit will be dropped as the vehicles move out of the RSU area. As update requests compete for bandwidth with other download requests, some data may become stale after an update is missed, degrading service quality. Therefore, we use the following metrics for scheduling vehicle–roadside data access compared with responsiveness and fairness, providing fresh data to more vehicles.

1. *Data quality.* Good data quality means data is not stale. Data become stale if a vehicle has the new version of the data but fails to upload it before the vehicle

moves out of the RSU range. The staleness of the data will degrade the data quality for the download service. In this chapter, we use the percentage of fresh data access to represent the data quality of the system. Therefore, a good scheduling scheme should update data in time and try to avoid data staleness.

2. *Service ratio.* A good scheduling scheme should serve as many requests as possible. The ratio of the number of requests served before the service deadline to the total number of arriving requests is the service ratio.

10.7.3 Roadside Unit Scheduling Schemes

Giving more bandwidth to download requests can provide a higher download service ratio, but a higher update drop ratio and hence low data quality. Therefore, achieving both high service ratio and good data quality is very difficult. If update requests get more bandwidth, the service ratio decreases.

There is always a trade-off between high service ratio and good data quality. Our focus now switches to improving the service ratio. The primary goal of a scheduling scheme is to serve as many requests as possible. We identify two parameters that can be used for scheduling vehicle–roadside data access:

1. *Deadline.* The request is not useful and should be dropped if a request cannot be served before its deadline. The request with an earlier deadline is more urgent than the request with a later deadline.
2. *DataSize.* Usually, vehicles can communicate with the RSU at the same data transmission rate. The data size decides how long the service will last.

Three naive schemes for roadside unit scheduling are as follows:

1. *First Deadline First* (*FDF*). In this scheme, the request with the most urgency will be served first.
2. *Smallest DataSize First* (*SDF*). In this scheme, the data with a small size will be served first.
3. *First Come First Serve* (*FCFS*). In this scheme, the request with the earliest arrival time will be served first.

The service ratios under these three naive scheduling schemes are compared in Figure 10.4. The interarrival time of the requests is determined by the percentage of vehicles that will issue service requests, which is varied along the x-axis. As shown in the figure, when the request arrival rate is low, FDF outperforms FCFS and SDF. This is because, when the workload is low, the deadline factor has more impact on the performance.

After the urgent requests are served, other pending requests can still have the opportunity to get services. However, when the request arrival rate increases, the service ratio of FDF drops quickly while SDF performs relatively better. Because the system can always find short requests for service, SDF can still keep a higher service ratio. FCFS does not take any deadline or data size factors into account when making scheduling decisions. It has the worst performance.

Data size and request deadlines are not considered in FCFS. FDF gives the highest priority to the most urgent requests while neglecting the service time spent on those

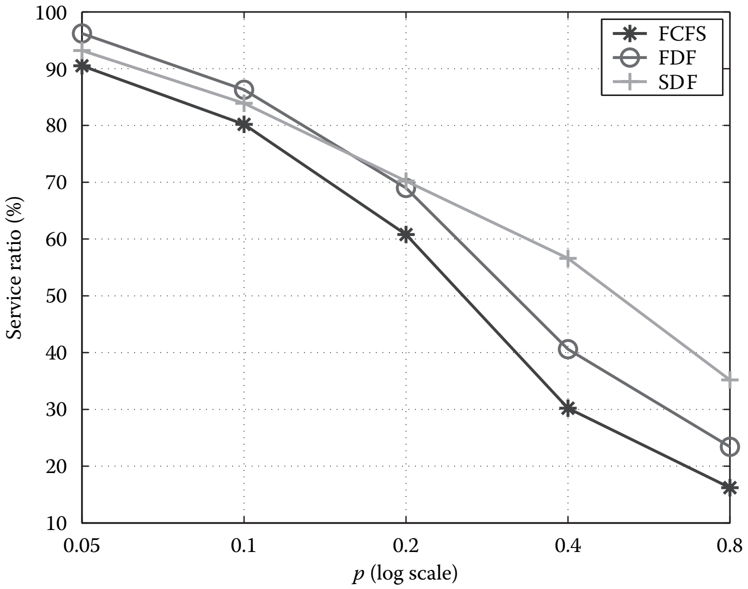

FIGURE 10.4 Service ratio for FCFS, FDF, and SDF schemes. (From Zhang, Y. et al., On Scheduling Vehicle–Roadside Data Access, in Proceedings of ACM VANET, 2007.)

data items. SDF takes the data size into account but ignores the request urgency. It is clearly shown in the figure that FDF and SDF can only achieve good performance for certain workloads.

This motivates the integration of the deadline and data size to improve the performance of scheduling. None of them can provide a good scheduling as a result. $\mathcal{D}*\mathcal{S}$[30] considers both data size and deadlines when scheduling vehicle–roadside data access. From the above observations, there are two principles are:

1. Given two requests with the same deadline, the one asking for a small size of data should be served first.
2. Given two requests asking for data with same size, the one with the earlier deadline should be served first.

Each request is given a service value based on its deadline and data size, called *DS_value*, as its service priority weight:

$$DS\_value = (Deadline - CurrentClock) * DataSize$$

In this equation, the deadline and data size factors are multiplied because these two factors have different measurement scales and/or units. With product, different metrologies will not impose any negative effect on the comparison of two *DS_values*. At each scheduling time, the $\mathcal{D}*\mathcal{S}$ scheme always serves the requests with the minimum *DS_value*.

10.8 Delay-Tolerant Routing in VANETs

Although most of the existing work on vehicle networks is limited to 1-hop or short-range multihop communication, vehicular delay-tolerant networks are useful to other scenarios. For example, without Internet connection, a moving vehicle may want to query a data center ten miles away through a VANET. The widely deployed wireless LANs or infostations[36,37] can also be considered.

Vehicle delay-tolerant networks have many applications, such as delivering advertisements and announcements regarding sale information or remaining stocks at a department store. Information such as the available parking spaces in a parking lot, the meeting schedule at a conference room, and the estimated bus arrival time at a bus stop can also be delivered by vehicle delay-tolerant networks.

For the limited transmission range, only clients around the access point can directly receive the data. However, this data may be beneficial to people in moving vehicles far away, as people driving may want to query several department stores to decide where to go. A driver may query the traffic cameras or parking lot information to make a better travel plan. A passenger on a bus may query several bus stops to choose the best stop for bus transfer. All these queries may be issued miles away from the broadcast site. With a vehicular delay-tolerant network, the requester can send the query to the broadcast site and get a reply from it. In these applications, the users can tolerate up to a minute of delay as long as the reply eventually returns.

The problem of efficient data delivery in vehicular delay-tolerant networks is studied in this section. Specifically, when a vehicle issues a delay-tolerant data query to some fixed site, we must know how to efficiently route the packet to that site and receive the reply with a reasonable delay. We will present a vehicle-assisted data delivery (VADD)[4] based on the idea of carry and forward.[38]

Some of the carry-and-forwarding approaches either pose too much control or no control at all on mobility, and hence are not suitable for vehicular networks. They include the ones proposed for delay-tolerant network.[14,22,38,39] In contrast, VADD makes use of predictable vehicle mobility, which is limited by the traffic pattern and road layout. For example, the driving speed is regulated by the speed limit and the traffic density of the road, the driving direction is predictable based on the road pattern, and the acceleration is bounded by the engine speed. VADD exploits the vehicle mobility pattern to better assist data delivery.

10.8.1 The VADD Protocol

In the model assumed by the VADD protocol, vehicles communicate with each other through a short-range wireless channel, and vehicles can find their neighbors through beacon messages. The packet delivery information such as source ID, source location, packet generation time, destination location, expiration time, and so on, are specified by the data source and placed in the packet header. A vehicle knows its location by triangulation or through a GPS device, which is already popular in new cars and will be common in the future.

Geographical information is also assumed to be available in the vehicles. Vehicles are equipped with preloaded digital maps, which provide street-level maps and traffic statistics such as traffic density and vehicle speed on roads at different times of the day. Such digital maps have already been commercialized. The latest one is developed by Map Mechanics,[40] and includes road speed data and an indication of the relative density of vehicles on each road. Yahoo! is also working on integrating traffic statistics in its new product called SmartView,[41] where real traffic reports of major U.S. cities are available.

It is expected that more detailed traffic statistics will be integrated into digital maps in the near future. The cost of setting up such a vehicular network can be justified by its application to many road safety and commercial applications,[1–3] which are not limited to the proposed delay-tolerant data-delivery applications.

The most important issue is to select a forwarding path with the smallest packet delivery delay. VADD is based on the idea of carry and forward. Although geographical forwarding approaches such as GPSR,[42] which always chooses the next hop closer to the destination, are very efficient for data delivery in ad hoc networks, they may not be suitable for sparsely connected vehicular networks.

Suppose a driver approaches intersection I_a and he wants to send a request to the coffee shop (to reserve a sandwich) at the corner of intersection I_b, as shown in Figure 10.5. To forward the request through $I_a \rightarrow I_c$, $I_c \rightarrow I_d$, $I_d \rightarrow I_b$ would be faster than forwarding through $I_a \rightarrow I_b$, even though the latter provides a geographically shortest-possible path. The reason is that, in the case of disconnection, the packet has to be carried by the vehicle, whose moving speed is significantly slower than the wireless communication. In sparsely connected networks, vehicles should try to make use of the wireless communication channel, and resort to vehicles with faster speed. Thus, VADD follows the following basic principles:

1. If the packet has to be carried through certain roads, the road with higher speed should be chosen.

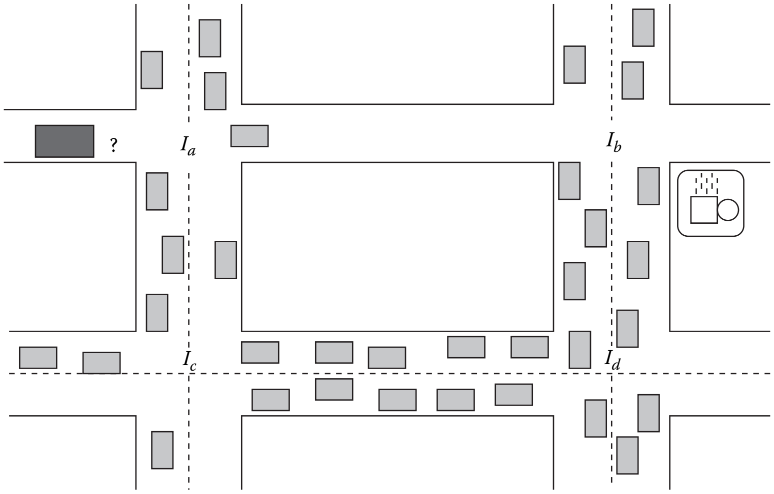

FIGURE 10.5 Find a path to the coffee shop.

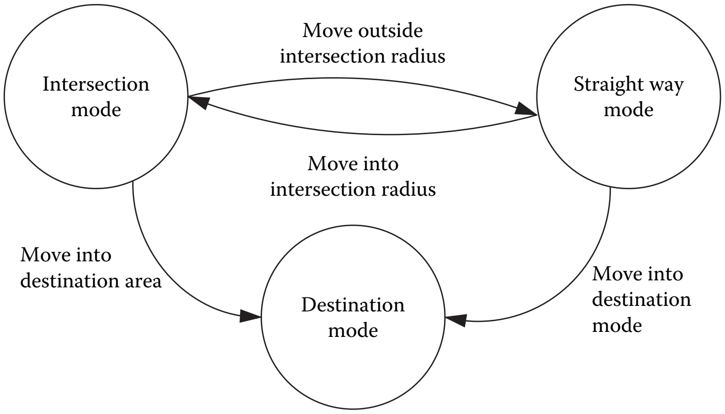

FIGURE 10.6 The transition mode in VADD.

2. Transmit through wireless channels as much as possible.
3. Owing to the unpredictable nature of VANETs, the packet cannot be expected to be successfully routed along the precomputed optimal path, so dynamic path selection should continuously be executed throughout the packet-forwarding process.

VADD has three packet modes (Figure 10.6): Intersection, Straight Way, and Destination, based on the location of the packet carrier (i.e., the vehicle that carries the packet.) By switching between these packet modes, the packet carrier takes the best packet-forwarding path. Among the three modes, the Intersection mode is the most critical and complicated one, because vehicles have more choices at the intersection.

10.9 Data Dissemination in VANETs

Data dissemination protocols[43,44] have been proposed to disseminate information about traffic, obstacles, and hazards on the roads. Similar applications such as real-time video streaming between vehicles have been studied.[45] A conventional way to report accidents or traffic conditions is to use certain infrastructures such as roadside traffic sensors reporting data to a central database, or cellular wireless communication between vehicles and a monitoring center. The problem with this design is the expensive deployment. In addition, these infrastructure-based networks are not scalable due to their centralized nature. VANETs, as an alternative to infrastructure-based vehicle networks, are constructed on-the-fly and do not require any investment besides the wireless network interfaces that will be a standard feature in the next generation of vehicles.

How to exchange traffic information among vehicles in a scalable fashion in VANETs is an interesting but challenging problem that has to be solved. Solutions to this problem can be categorized into two main mechanisms: a flooding-based approach and a dissemination-based approach. In the flooding mechanism, each individual vehicle periodically broadcasts information about itself. Every time a vehicle receives a broadcast message, it stores it and immediately forwards it by rebroadcasting the message. This mechanism is clearly not scalable due to the large volume of messages flooded over the network, especially in high-traffic-density scenarios. The flooding-based mechanism can be further divided into three categories: priority-based approaches, distance-based approaches, and geocast approaches. On the other hand, in the dissemination mechanism, each vehicle broadcasts information about itself and the other vehicles it knows about. Each time a vehicle receives information broadcasted by another vehicle, it updates its stored information to the next broadcast period, at which time it broadcasts its updated information. The dissemination mechanism is scalable, because the number of broadcast messages is limited, and they do not flood the network. The dissemination-based mechanism can be further divided into two categories: approaches utilizing the bidirectional mobility of vehicles and forwarding-based approaches. We can see the classification of mechanisms of multicast/broadcast in VANETs in Figure 10.7.

10.9.1 Flooding-Based Mechanisms

A number of safety applications require communications to a group of vehicles, not just pairwise communications supported by unicast protocols. Safety applications require propagation of information to a large number of nodes quickly and reliably. Flooding is the most common approach for broadcasting without explicit neighbor information. However, flooding is known to be inefficient due to the so-called broadcast storm problem, especially in networks with high node density. Most existing flooding-based information dissemination approaches in VANETs aim to achieve a high message delivery ratio by avoiding contention and collision caused by the broadcast storm phenomena.[43,46,47]

10.9.1.1 Priority-Based Approach

In Ref. 48, the authors study how broadcast performance scales in VANETs and propose a priority-based broadcast scheme that gives higher priority to nodes that need to transmit time-critical messages. The proposed algorithm categorizes nodes in the network

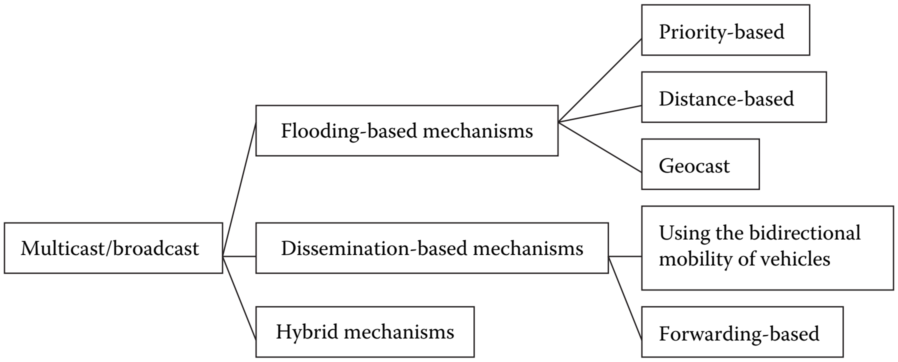

FIGURE 10.7 The classification of multicast/broadcast mechanisms.

into multiple classes with different priorities and schedules the packet transmission accordingly. Although this technique is not designed to solve the broadcast storm problem, it can indirectly mitigate the severity of the storm by allowing nodes with higher priority to access the channel as quickly as possible.

10.9.1.2 Distance-Based Approach

Reference 43 proposed an efficient 802.11-based urban multihop broadcast protocol (UMB) where only the furthest vehicle from the transmitter rebroadcasts the packet such that broadcast redundancy is suppressed.

In Ref. 46, three distance-based mechanisms were proposed:

1. Weighted p-persistence broadcasting
2. Slotted 1-persistence broadcasting
3. Slotted p-persistence broadcasting

The basic broadcast techniques follow either a p-persistence rule or a 1-persistence ($p = 1$) rule. Although the overhead is excessive, most routing protocols designed for multihop ad hoc wireless networks follow the brute-force 1-persistence flooding rule where all nodes in the network rebroadcast the packet with probability 1. On the other hand, the gossip-based approach follows the p-persistence rule where each node rebroadcasts with a predetermined probability p. This method is also referred to as probabilistic flooding.[49]

In weighted p-persistence broadcasting, upon receiving a packet from node i, node j checks the packet ID and rebroadcasts with probability p_{ij} if it is the first time that node j receives the packet; otherwise, the packet is discarded by node j. The forwarding probability, p_{ij}, can be calculated on a per-packet basis using the following expression, $p_{ij} = D_{id}/R$ where D_{ij} is defined as the relative distance between node i and j, and R is the average transmission range. Unlike the p-persistence scheme, the weighted p-persistence broadcasting assigns nodes that are farther away from the broadcaster higher probability given that the GPS information is available and accessible from the header of a packet. The weighted p-persistence approach is illustrated in Figure 10.8a.

In slotted 1-persistence broadcasting, upon receiving a packet, a node checks the packet ID and rebroadcasts with probability 1 at the assigned timeslot, if it is the first time it receives the packet and it has not received any duplicate packets before its assigned timeslot $T_{S_{ij}}$; otherwise, the packet is discarded. Denoting D_{ij} as the relative distance between node i and j, R as the average transmission range, and N_s the predetermined number of slots, $T_{S_{ij}}$ can be calculated as $T_{S_{ij}} = S_{ij} \times \tau$ where τ is the estimated 1-hop delay, which includes the propagation delay and the medium access delay, and S_{ij} is the assigned slot number which is defined as $S_{ij} = N_s \times \left(1 - \lceil \min(D_{ij}, R)/R \rceil\right)$. The timeslot method follows the same logic as the weighted p-persistence scheme. However, each node uses the GPS information to calculate the waiting time to retransmit instead of calculating the reforwarding probability. For example, in Figure 10.8b, the broadcast coverage is spatially divided into four regions and the nodes located in the farthest region will be assigned a shorter waiting time. Therefore, a node takes on the smallest D_{ij} value if it receives duplicate packets from more than one sender. Similar to the p-persistence scheme, this method needs the transmission range information so as to agree on a certain value of slot size or the number of slots. Note that N_s is a design parameter that needs to be carefully selected.

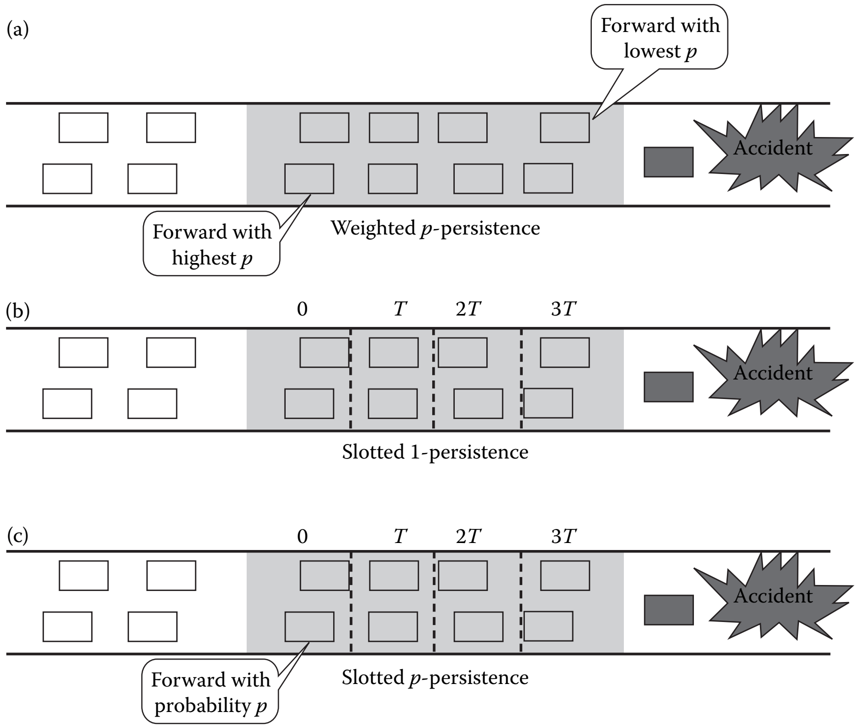

FIGURE 10.8 An example of the broadcast suppression technique.

In slotted p-persistence broadcasting, upon receiving a packet, a node checks the packet ID and rebroadcasts with the predetermined probability p at the assigned timeslot $T_{S_{ij}}$, if it is the first time that it receives the packet and it has not received any duplicate packets before its assigned timeslot; otherwise, the packet is discarded by it. Figure 10.8c illustrates the concept of the slotted p-persistence with four slots.

Reference 46 asserted that the slotted p-persistence scheme can substantially reduce the packet loss ratio at the expense of a slight increase in total delay and reduced penetration rate.

10.9.1.3 Flooding Geographically Defined Information

In many safety applications, vehicle safety alarms are required to be sent to all vehicles within a specific area where protocols for flooding geographically defined information are needed. Geocast is a variation of conventional multicast, which specifies the destination as a geographic position rather than a specific node or multicast addresses. The multicast group (or geocast group) is implicitly defined as the set of nodes within a specified area, which is different from the conventional multicast schemes. That is, a node automatically becomes a member of the corresponding geocast group at a given time if it is within the geocast region at that time.

The IVG protocol proposed in Ref. 50 addresses how to broadcast alarm messages only to vehicles approaching areas of a given accident. An alarm message received is discarded if it is not relevant. Otherwise, it is rebroadcasted if this message is still relevant after a deferred period of time.

10.9.2 Dissemination-Based Mechanisms

Compared to the flooding-based approaches, dissemination-based mechanisms are more scalable because the number of broadcast messages is limited, and they do not flood the network. The dissemination mechanism can either broadcast information to vehicles in all directions, or perform a directed broadcast restricting information about a vehicle to vehicles behind it.

10.9.2.1 Data Dissemination Considering the Bidirectional Mobility of Vehicles

Reference 51 presents a formal model of data dissemination in VANETs and studies how the performance of data dissemination is affected by VANET characteristics, especially the bidirectional mobility on well-defined paths. The analysis as well as simulation results show that dissemination using only vehicles in the opposite direction significantly increases the data dissemination performance.

Without loss of generality, vehicles are assumed to move on bidirectional straight roads with multiple lanes in each direction, as shown in Figure 10.9. It is assumed that a vehicle on the road moves either to the *East* as shown in the lower part of the road

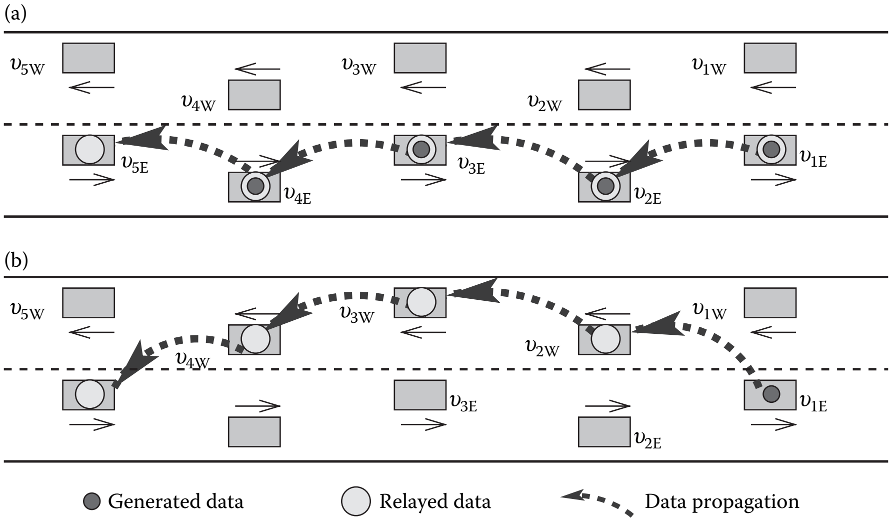

FIGURE 10.9 Dissemination models: (a) the same-dir dissemination model, and (b) the opp-dir dissemination model.

in Figure 10.9 (e.g., v_{1E} and v_{2E}), or to the *West* as shown in the upper part of the road (e.g., v_{1W} and v_{2W} in Figure 10.9). S_E and S_W are the average speeds for *East* and *West* directions, respectively. All transmissions are assumed to be omnidirectional with communication range R. Each vehicle in the model is assumed to be concerned about the road information ahead of it. Information should propagate backwards with respect to the vehicle's direction (i.e., propagates in the opposite direction) in order to accomplish this assumption. Assume that vehicles broadcast data packets periodically every B seconds. For the sake of simplicity, consider the propagation of information about vehicles moving *East* where the direction of propagation is from the east to the west.

There are two types of broadcasted data: generated data and relayed data. Generated data, denoted by a small red circle in Figure 10.9, is the vehicle's own data (e.g., ID, speed, and location) and it is updated every new broadcast period. Relayed data, denoted by a large yellow circle, is the stored data about the other vehicles ahead, and is propagated backward within every broadcast period. Three dissemination/propagation models are considered: *same-dir*, *opp-dir*, and *bi-dir*. In the *same-dir* model, every vehicle periodically broadcasts both the store-relayed data and its generated data in the same data packet. When a vehicle broadcasts a data packet, only vehicles moving in the same direction are involved in the propagation of this packet. More specifically, when v_1 broadcasts a data packet, v_2 will propagate it later if and only if the following all apply: **Q6**

1. v_2 is within the transmission range of v_1
2. v_1 and v_2 are moving in the same direction (i.e., *East*)
3. v_1 is in front of v_2 with respect to their directions (i.e., v_1 is located east to v_2)

Figure 10.9a is an example of how information is propagated from vehicle v_{1E} to vehicle v_{5E}, both moving in the *East* direction in the *same-dir* model. Note that no vehicle from the opposite direction is involved in the dissemination in this model.

On the other hand, in the *opp-dir* model, relayed data and generated data are not broadcast together. Instead, vehicles in the same direction (i.e., *East*) only broadcast their generated data. These generated data are aggregated and propagated backwards by the vehicles in the opposite direction (i.e., *West*). When v_1 broadcasts a packet (i.e., relayed data in the case moving *West*, or generated data in the case moving *East*), v_2 will operate according to the following rules, given that it is within the the transmission range of v_1:

1. If v_1 and v_2 are moving *East*, v_2 will accept the packet if v_1 is located east of v_2. This is the case when v_1 broadcasts its generated data.
2. If v_1 and v_2 are moving *West*, v_2 will accept the packet if v_2 is located west of v_1. This is the case when v_1 relays a packet.
3. If v_1 is moving *East* (or *West*) and v_2 is moving *West* (or *East*), v_2 will accept the packet regardless of the relative position of the vehicles.

The first rule guarantees a fast delivery of the newly generated data to all the vehicles within one hop of the source vehicle. Figure 10.9b is an example of how information is propagated from v_{1E} to v_{5E} in the *opp-dir* model.

The *bi-dir* model combines both the *same-dir* and the *opp-dir* models. In this model, vehicles in the same direction(i.e., *East*) are involved in the propagation of generated and relayed data while vehicles in the opposite direction (i.e., *West*) only propagate relayed data. Information in this model is propagated by vehicles moving in both the same and the opposite directions, which is different from the other mechanisms.

Analysis and simulation show that the performance of the data dissemination model relies on the traffic densities in both directions of the road. When traffic in the opposite direction (e.g., *West*) is not sparse, the *opp-dir* model is more efficient than both the *bi-dir* and the *same-dir* models in terms of latency, network utilization, and average error. Although the *bi-dir* model has better knowledge than the *opp-dir* model in this network configuration, this better knowledge comes with the cost of lower utilization rates, higher latency, and lower accuracy. This indicates that the *opp-dir* model is the most efficient data-dissemination model in terms of scalability, accuracy, and efficiency. However, the *bi-dir* model outperforms both the *opp-dir* and the *same-dir* models when traffic in the opposite direction is sparse.

10.9.2.2 Forwarding-Based Data Dissemination Protocols

Several forwarding-based protocols for data dissemination have been proposed recently. An opportunistic forwarding approach is proposed in Ref. 52. It is asserted that the motion of vehicles on a highway can contribute to successful message delivery, provided that messages can be disseminated in a store-carry-forward fashion. Reference 53 proposes a trajectory-based forwarding scheme. Reference 54 proposes MDDV, a combination of opportunistic forwarding and trajectory-based forwarding, which specifically addresses vehicle mobility. MDDV, a mobility-centric approach for data dissemination in vehicular networks, is designed to operate reliably and efficiently in spite of the highly dynamic, partitioned nature of these networks. MDDV is designed to exploit vehicle mobility for data dissemination, and combines the idea of geographical forwarding, opportunistic forwarding, and trajectory-based forwarding.

A forwarding trajectory is specified as a path from the source to the destination region. The road network can be abstracted as a directed graph where nodes represent intersections and edges represent road segments. One of the MDDV objectives is to deliver messages to their destination regions with low delay. Taking the path with the shortest distance from the source to the destination region would be a naive approach in that information propagation along a road depends largely on the vehicle traffic on it, for example, vehicle density besides the distance between the source–destination pair. **Q7** A short road distance does not necessarily result in short information propagation delay. High vehicle density often guarantees fast information propagation. Therefore both the traffic condition and the road distance must be taken into consideration. However, vehicle traffic conditions vary from one road segment to another and change over time. The number of lanes gives some indication of the expected vehicle traffic. $d(A, B)$ is defined as the "dissemination length" of a road segment from road node A to B, which takes into consideration the static road information. Denoting $r(A, B)$ as the road length between A and B, i/j as the number of lanes from A/B to B/A, the following heuristic formula is used: $d(A, B) = r(A, B)(m - (m - 1)(i^p + cj^p))$ where $0 < c < 1$. The dissemination length of a road segment is used as the weight for the corresponding link in the abstracted road graph. MDDV uses a forwarding trajectory that is specified as the directed path with the smallest sum of weights from the source to the destination region in the weighted road graph.

The dissemination process consists of two phases: the forwarding phase and propagation phase. In the forwarding phase, the message is forwarded along the forwarding trajectory to the destination region. The propagation phase begins and the message is propagated to each vehicle in an area centered on the destination region before the message time expires, once the message reaches the destination region. In order to deliver the message to the intended receivers before they enter the destination region in order to reduce delay, this area covers the destination region and is usually larger.

At first, Ref. 54 assumes that each vehicle has perfect knowledge concerning the global status of the data dissemination. During the forwarding phase, the message holder closest to the destination region along the forwarding trajectory is called the "message head." The vehicle taking the role of the message head may change over time as the message propagates or vehicles move. With perfect knowledge, every vehicle knows the message head vehicle in real time. Only the message head tries to pass the message to other vehicles that may be closer to the destination region. During the propagation phase, the message is propagated to vehicles without the message in the specified area.

Owing to the lack of perfect knowledge for participating vehicles, the above ideal scenario cannot be implemented. In practice, individual vehicles have no idea about which vehicle is the message head in real time. For instance, as illustrated in Figure 10.10, on a two-way traffic road, the current message head is vehicle 1. In Figure 10.10a, vehicle 1 may run out of the trajectory or may become inoperative; vehicle 2, the immediate follower, may not be aware of this because the network is partitioned. In Figure 10.10b, vehicle 1 is moving away from the destination region (note that the road is bidirectional). Once vehicle 1 passes vehicle 2, vehicle 2 should become the new message head. However, vehicle 2 does not know this unless it receives an explicit notification from vehicle 1. With the assumption that vehicles do not know the location of others, this is difficult to do. In both cases, the message is lost. To address this problem, a group

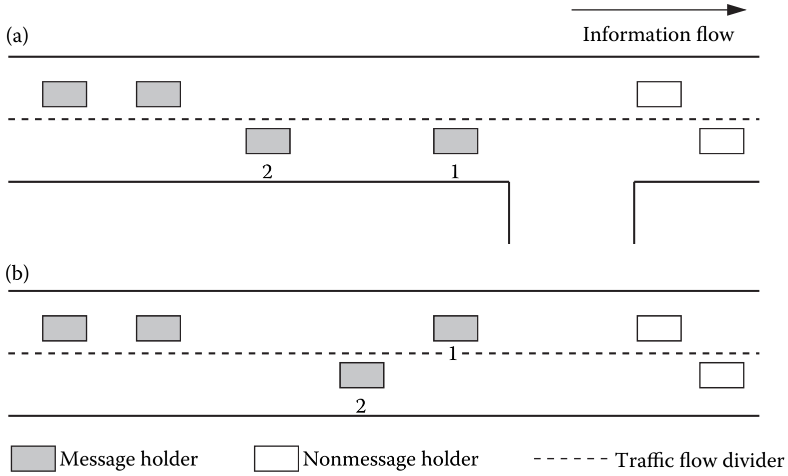

FIGURE 10.10 An example of lack of perfect knowledge.

of vehicles near the real message head can actively forward the message instead of the message head vehicle only. The group membership changes as the actual message head moves toward the destination region. There is a trade-off between delivery reliability and message overhead: larger groups mean higher delivery reliability but higher message overhead, too. Vehicles have to locally determine their own actions based on their approximate knowledge of the global message dissemination status.

Vehicles need to have some information regarding the message dissemination status in order to realize the approximation. Specifically, information concerning the message head is required. However, the message dissemination status changes over time. Vehicles can only expect approximate knowledge, at best. Also a vehicle's knowledge must be updated constantly. A convenient way to exchange such information is to place it in the message. As the message is propagated among vehicles, so does the message dissemination status information. Too much information in the message is cumbersome and expensive, however. To this end, a small amount of data, the message head location and its generation time, called the *message head pair*, is inserted into the message. Every holder of a message maintains a message record containing the message head pair along with other information concerning this message. The message head pair provides the best knowledge of a message holder regarding the message head location.

The actual message head can move either away from or towards the destination region along the forwarding trajectory within a short period of time. But it should move towards the destination region in the long run (because the message head vehicle may change). For simplicity, it is required that the message head location installed by a message holder never moves backward, which means that a message holder can only install a new message head location closer to the destination region than the one currently installed.

To reduce the publication and dissemination of false information, only some vehicles are allowed to generate the message head pair. A message holder is allowed to publish its current location as the message head location if it believes it may be the real message head with some probability. In this sense, a message holder may assume either one of two roles: the message head candidate and nonmessage head candidate. Only a message head candidate can actively publish its current location as the message head location and a nonmessage head candidate can only learn from received messages.

There are rules for a message holder to transit between a message head candidate and nonmessage head candidate. Suppose the current time is t_c, a vehicle's current location is l_c, and a vehicle's installed message head pair is $\langle l, t \rangle$, where l is the message head location and t is the generation time:

1. *Nonmessage head candidate → message head candidate.* During the forwarding phase, one important observation is that a vehicle passing its installed message location a shorter period after the generation time is more likely to be the message head, because after a long period the message may have already been forwarded far away toward the destination region along the trajectory. Thus a nonmessage head candidate becomes a message head candidate if it passes its installed message head location toward the destination region before $t + T_1$, where T_1 is a system parameter. During the propagation phase, message holders moving into the destination region assume the role of the message head candidate.

2. *Message head candidate → nonmessage head candidate.* During the forwarding phase, there are two transition rules: (1) if the message head candidate leaves the trajectory or moves away from the destination region along the trajectory, it becomes a nonmessage head candidate; (2) if a message head candidate moves toward the destination region along the trajectory, it stays as a message head candidate until it receives the same message with another message head pair $\langle l_n, t_n \rangle$ where l_n is closer to the destination region than l_c. During the propagation phase, a message head candidate becomes a nonmessage head candidate once it moves out of the destination region.

A message holder updates its installed message head pair with the information from received messages. Two messages differing only in the message head pair are two versions of the same message. One message version with message head pair $\langle l_i, t_i \rangle$ is said to be newer than another message version with message head pair $\langle l_j, t_j \rangle$ if l_i is closer to the destination region than l_j; or $l_i = l_j$ but $t_i > t_j$. A vehicle always updates its installed message head pair with the newer received information. Therefore obsolete/false installations can be eliminated through data exchange.

The data exchange algorithm is defined as the following:

1. *Forwarding phase.* A message holder can be in either one of two dissemination states: the active state and passive state, or not eligible to transmit at all. A message holder in the active state runs the full protocol to actively propagate the message while a message holder in the passive state only transmits the message if it hears some older message version. The active propagation can help populate the message, move the message closer to the destination region, or update dissemination status. The passive updating serves to eliminate false/obsolete information only. Given a message holder's installed message head pair $\langle l, t \rangle$, its current location l_c and the current time t_c, it is in the active state if $t_c < t + T_2$ and l_c is within the distance L_2 from l, and otherwise it is in the passive state if $t_c < t + T_3$ and l_c is within the distance L_3 from l, while $T_2 < T_3$ and $L_2 < L_3$. Otherwise, the message will not be transmitted under any circumstance. T_2, T_3, L_2, and L_3 are system parameters. In this way, the active data propagation is initiated by the fresh generation of a message head pair and is constrained near the message head location (through both geographical and temporal constraints). Data propagation caused by obsolete/false information will eventually stop when the time expires or it is suppressed by updates.
2. *Propagation phase.* A message holder can either be in the active state or not eligible to transmit. A message holder in the active state runs the full protocol. The active propagation serves to deliver the message to intended receivers. Using the same notations as before, a message holder is in the active state if $t_c < t + T_2$ and l_c is within the distance L_2 from l. Every vehicle inside the destination region publishes its own location as the message head location. Therefore this data exchange mechanism limits the active propagation in a region centered on the destination region.

It is important for an opportunistic forwarding mechanism to determine when to store/drop a message. The design decision can influence memory usage, message overhead, and delivery reliability. The decision to store/drop messages can be based on a vehicle's knowledge of its future movement trajectory. For example, a message holder may decide to drop a message if it knows that continually holding the message can no longer contribute to suppress unnecessary message transmissions based on its future movement trajectory, given that vehicles are aware of their own near-future movement trajectory. In MDDV, memory buffers are assumed to be free from limit such that each vehicle stores whatever it overhears. A message is dropped by a vehicle when the vehicle leaves the active state during the propagation phase, leaves the passive state during the forwarding phase, or the message expiration time elapses.

10.9.3 Hybrid Mechanisms

Flooding-based data dissemination mechanisms are unscalable due to the large amount of contention and collision, especially in dense networks. On the other hand, dissemination-based mechanisms are not suited for delay-sensitive safety message dissemination, albeit scalability is achieved. Hence, hybrid mechanisms that combine the strengths of each are proposed. Reference 55 proposes an approach (called Directional Propagation Protocol, or simply, DPP) using clusters of connected vehicles where flooding-based data dissemination mechanisms are used in a cluster and dissemination-based mechanisms are used among clusters.

DPP uses the directionality of data and vehicles for information propagation. DPP comprises three components: a Custody Transfer Protocol (CTP), an Inter-Cluster Routing Protocol, and an Intra-Cluster Routing Protocol. In order to overcome the lack of an end-to-end path between source and destination, the Custody Transfer Protocol is introduced which is derived from delay-tolerant networking concepts. On the one hand, the Inter-Cluster Routing Protocol controls the message exchange between nodes within a cluster. On the other hand, the communication between clusters is governed by the Intra-Cluster Routing Protocol. As illustrated in Figure 10.11, interconnected blocks of vehicles can be formed by vehicles traveling towards the same direction. Gaps are allowed between consecutive blocks. The traffic density has a significant impact on the cardinality of each block. For example, a long continuous block can be formed under dense conditions, while under sparse conditions, the cardinality of each block could be one.

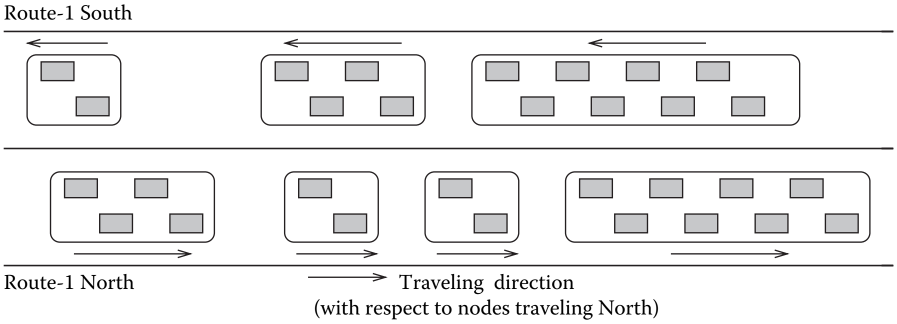

FIGURE 10.11 An example of blocks of vehicles.

Additionally, vehicles that are within range *R* and maintain connectivity for a minimum time *t* are said to be part of a cluster. Thus, a block may comprise several clusters.

Under sparse traffic conditions, gaps between blocks are frequent and network partitions are common, which prevents an end-to-end path between source and destination. Accordingly, the speed of the vehicle that carries the message may influence the data dissemination performance. Under dense traffic conditions, an end-to-end path between source and destination exists with high probability where the data dissemination performance is mainly determined by contentions and collisions.

The effects of speed differentials within the cluster are not considered as the faster vehicles will leave one cluster and join another as they progress on the road. Also, there are intersections on a highway where vehicles may join or leave the clusters. Once a cluster becomes very large, the cluster is split to better manage intracluster traffic.

Each cluster has a header and a trailer, located at the front and rear of each cluster, entrusted with the task of communicating with other clusters. A node at the head or tail of the cluster will elect itself as the header or trailer for our protocol. (Node election is not covered here.) This limits congestion caused by the large number of participating nodes. The remaining nodes in the cluster, nodes that are not header or trailer, are described as intermediate nodes. Within a cluster, communicated messages are shared with all nodes to both facilitate header/trailer replacement and general awareness of disseminated messages.

The intermediate nodes retain a passive role of receiving messages and acknowledgments from opposing blocks and forwarding them to the header or trailer sharing the information within the cluster. Similarly, messages originating from intermediate nodes are immediately routed to header or trailer depending upon the direction in which information needs to propagate. Any duplicate messages received at any of the nodes are dropped. End-to-end path formation can be assumed to be taking place within a cluster.

In most message-passing schemes, a message is buffered until an acknowledgment from the destination is received. However, due to network fragmentation in a VANET and the resultant lack of continuous end-to-end connectivity at any given instant, the message can require buffering for an indeterminate amount of time. The result translates to the requirement for large buffer sizes or dropped messages and difficulty in exchanging acknowledgments. For applications that do not require continuous end-to-end connectivity, a store-and-forward approach can be used.

With the custody transfer mechanism, a message is buffered for retransmission from the originating cluster until it receives an acknowledgment from the next-hop cluster. The custody is implicitly transferred to another cluster that is in front along the direction of propagation and is logically the next hop in terms of the message path. The traffic in the opposing direction acts as a bridge but is never given custody of the message. The custody is not released until an acknowledgment is received from the cluster in front. Once the message reaches the next-hop cluster, it has custody of the message and the responsibility for further relaying the message is vested with this cluster. The custody of the message may be accepted or denied by a cluster by virtue of it being unable to satisfy the requirements of the message.

The propagation is called *reverse propagation* if the data are headed in a direction opposite to the direction of motion of the vehicles and *forward propagation* if data are

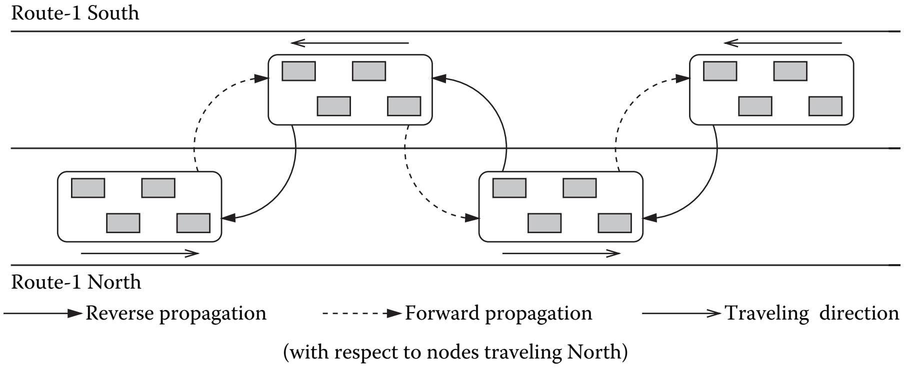

FIGURE 10.12 An example of forward propagation and reverse propagation.

headed along the direction of motion of the vehicles. In forward propagation, as illustrated in Figure 10.12, the vehicle is assumed to be traveling along the N direction and the data are also to be propagated in the N direction. The data can travel at a minimum rate of the speed of the vehicle because the data are traveling along with the vehicle. The data are propagated to the header of the cluster. The header now tries to propagate the data further along the N direction, trying to communicate with other clusters located ahead of this cluster. If the clusters are partitioned, the header attempts to use the clusters along the S direction, which may overlap with other clusters along the N direction to bridge this partition. Thus, the data are propagated to nodes traveling along the N direction that are otherwise partitioned from each other, by using clusters along the S direction. This temporary path occurs due to opportunistic contact with nodes in the overlapping clusters. Once the data are forwarded to the next hop and an acknowledgment is received, the custody is transferred to that cluster. The entire process is repeated until the data reaches its required destination. The reverse propagation scheme can be modeled as an extension of the forward propagation scheme.

10.10 Conclusion

In this chapter, we studied the problem of efficient data delivery (unicast) and dissemination (multicast) in delay-tolerant vehicle networks.

First, we described the concepts of delay-tolerant networks (DTNs) including the characteristics of these networks, store-carry-forward routing protocols. Because VANETs are a special type of DTN, which result from its unique mobility pattern, we further studied the vehicle traffic model. Delay-tolerant vehicle networks can be either infrastructure-based or infrastructure-free. We studied the role of roadside units in DTN routing in infrastructure-based vehicle networks. We also studied routing protocols such as VADD in infrastructure-free vehicle networks. As an emphasis, we studied the problem of data dissemination in VANETs where flood-based mechanisms, dissemination-based mechanisms, and hybrid mechanisms were visited.

We believe that routing and data dissemination are still challenging problems due to the unique characteristics of vehicle networks, such as high node mobility, high probability of network disconnection, and partition and lane-based node move pattern. This chapter summarizes some, but not all of the important findings in the vehicle networks community regarding routing and data dissemination. We hope our work lays the groundwork for future study in this area.

References

1. Xu, Q., Mark, T., Ko, J., and Sengupta, R., Vehicle-to-Vehicle Safety Messaging in DSRC, in Proc. of ACM VANET, 2004. **Q8**
2. Yang, X., Liu, J., Zhao, F., and Vaidya, N., A Vehicle-to-Vehicle Communication Protocol for Cooperative Collision Warning, in Proc. of ACM MOBIQUITOUS, August 2004.
3. Yin, J. et al., Performance Evaluation of Safety Applications Over DSRC Vehicular Ad Hoc Networks, in Proc. of ACM VANET, 2004.
4. Zhao, J., and Cao, G., VADD: Vehicle-Assisted Data Delivery in Vehicular Ad Hoc Networks, in Proc. of IEEE INFOCOM, 2006.
5. Namboodiri, V., Agarwal, M., and Gao, L., A Study on the Feasibility of Mobile Gateways for Vehicular Ad-Hoc Networks, in Proc. of ACM VANET, 2004.
6. Clausen, T. and Jacquet, P., Optimized Link State Routing Protocol OLSR, in RFC 3626, IETF Network Working Group, 2003.
7. Fall, K., A Delay-tolerant Network Architecture for Challenged Internets, in Proc. of ACM SIGCOMM, 2003.
8. Cerf, V. et al., Delay Tolerant Network Architecture, available at http://www.ietf.org/rfc/rfc4838.txt, 2007.
9. Cardei, I., Liu, C., and Wu, J., Routing in Wireless Networks with Intermittent Connectivity, in Encyclopedia of Wireless and Mobile Communications, CRC Press, Taylor & Francis Group, 2007.
10. Jain, S., Fall, K., and Patra, R., Routing in a Delay Tolerant Network, in Proc. of ACM SIGCOMM, 2004.
11. Merugu, S., Ammar, M., and Zegura, E., Routing in Space and Time in Networks with Predictable Mobility, in GIT-CC-04-7, 2004.
12. Gnawali, O., Polyakov, M., Bose, P., and Govindan, R., Data Centric, in Proc. of IEEE Aerospace Conference, 2005.
13. Intanagonwiwat, C., Govindan, R., and Estrin, D., Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks, in Proc. of ACM MOBICOM, 2000.
14. Vahdat, A. and Becker, D., Epidemic Routing for Partially Connected Ad Hoc Networks, April 2000. **Q9**
15. Spyropoulos, T., Psounis, K., and Raghavendra, C.S., Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks, in Proc. ACM SIGCOMM Workshop on Delay-Tolerant Networking, 2005.
16. Shah, R.C., Roy, S., Jain, S., and Brunette, W., Data Mules: Modeling a Three-tier Architecture for Sparse Sensor Networks, in Proc. IEEE International Workshop on Sensor Network Protocols and Applications, 2003.

17. Lindgren, A., Doria, A., and SchelSn, O., Probabilistic Routing in Intermittently Connected Networks, in Poster of ACM MOBIHOC, 2003.
18. Burns, B., Brock, O., and Levine, B.N., Mv Routing and Capacity Building in Disruption Tolerant Networks, in Proc. IEEE INFOCOM, 2005.
19. Juang, P. et al., Energy-efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with Zebranet, in Proc. of ASPLOS, 2002.
20. Leguay, J., Friedman, T., and Conan, V., Evaluating Mobility Pattern Space Routing for DTNs, in Proc. of IEEE INFOCOM, 2006.
21. Ratnasamy, S., Francis, P., Handley, M., Karp, R., and Shenker, S., A Scalable Content Addressable Network, in Proc. of ACM SIGCOMM, 2001.
22. Zhao, W., Ammar, M., and Zegura, E., New directions: A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks, in Proc. of ACM MOBIHOC, 2004.
23. Zhao, W., Ammar, M., and Zegura, E., Controlling the Mobility of Multiple Data Transport Ferries in a Delay-Tolerant Network, in Proc. of IEEE INFOCOM, 2005.
24. Yang, J., Chen, Y., Ammar, M., and Lee, C., Ferry Replacement Protocols in Sparse MANET Message Ferrying Systems, in Proc. of IEEE WCNC, 2005.
25. Wisitpongphan, N., Bai, F., Mudalige, P., Sadekar, V., and Tonguz, O.K., On the Routing Problem in Disconnected Vehicular Networks, in Proc. of IEEE INFOCOM Minisymposia, 2007.
26. *Highway Capacity Manual 2000*, May 2000. **Q10**
27. Sahoo, P.K., Rao, S.K., and Kumar, V.M., A Study of Traffic Characteristics on Two Stretches of National Highway No 5, Indian Highways, Indian Road Congress, 1996.
28. Verdone, R., Multi-Hop R-Aloha for Inter-Vehicle Communication at Millimeter Waves, *IEEE Transactions on Vehicular Technology*, November 1997. **Q11**
29. Lott, M., Halmann, R., Schulz, E., and Radimirsch, M., Medium Access and Radio Resource Management for Ad Hoc Networks Based on UTRA TDD, in Poster of ACM MOBIHOC, 2001.
30. Zhang, Y., Zhao, J., and Cao, G., On Scheduling Vehicle-Roadside Data Access, in Proc. of ACM VANET, 2007.
31. Su, C. and Tassiulas, L., Broadcast Scheduling for Information Distribution, in Proc. of IEEE INFOCOM, 1997.
32. Aksoy, D. and Franklin, M., R × W: A Scheduling Approach for Large-scale On-demand Data Broadcast. *IEEE/ACM Transactions on Networking*, 7(6), 1999. **Q11**
33. Gandhi, R., Khuller, S., Kim, Y., and Wan, Y., Algorithms for Minimizing Response Time in Broadcast Scheduling, Algorithmica, 2004.
34. Acharya, S. and Muthukrishnan, S., Scheduling On-demand Broadcasts: New Metrics and Algorithms, in Proc. of ACM MOBICOM, 1998.
35. Wu, Y. and Cao, G., Stretch-Optimal Scheduling for On-demand Data Broadcasts, in Proc. of IEEE ICCCN, 2001.
36. Frenkiel, R.H., Badrinath, B.R., Borras, J., and Yates, R., The Infostations Challenge: (Balancing Cost and Uiquity in Delivering Wireless Data, *IEEE Personal Communications*), April 2000. **Q11**
37. Goodman, D., Borras, J., Mandayam, N., and Yates, R., INFOSTATIONS: A New System Model for Data and Messaging Services, in Proc. of IEEE VTC, May 1997.

38. Davis, J., Fagg, A., and Levine, B., Wearable Computers as Packet Transport Mechanisms in Highly-Partitioned Ad-hoc Networks, in Proc. of Fifth International Symposium on Wearable Computers, October 2001.
39. Li, Q. and Rus, D., Sending Messages to Mobile Users in Disconnected Ad-hoc Wireless Networks, in Proc. of ACM MOBICOM, 2000.
40. U.S. Census Bureau, Tiger, Tiger/line and tiger-related products. **Q9**
41. SmartView, available at http://maps.yahoo.com/smartview.php.
42. Karp, B. and Kung, H.T., GPSR: Greedy Perimeter Stateless Routing for Wireless Networks, in Proc. of ACM MOBICOM, 2000.
43. Korkmaz, G., Ekici, E., Ozguner, F., and Ozguner, U., Urban Multi-Hop Broadcast Protocol for Inter-Vehicle Communication Systems, in Proc. of ACM VANET, October 2004.
44. Xu, B., Ouksel, A., and Woflson, O., Opportunistic Resource Exchange in Inter-vehicle Ad Hoc Networks, in Proc. of IEEE MDM, 2004.
45. Ghandeharizadeh, S., Kapadia, S., and Krishnamachari, B., PAVAN: A Policy Framework for Content Availability in Vehicular Ad-hoc Networks, in Proc. of ACM VANET, October 2004.
46. Wisitpongphan, N. et al., Broadcast Storm Mitigation Techniques in Vehicuar Ad Hoc Wireless Networks, IEEE Wireless Communication Magazine, 2007. **Q11**
47. Korkmaz, G., Ekici, E., and Ozguner, F., An Efficient Fully Ad-hoc Multi-hop Broadcast Protocol for Inter-vehicular Communication Systems, in Proc. of IEEE ICC, 2006.
48. Torrent-Moreno, M., Jiang, D., and Hartenstein, H., Broadcast Reception Rates and Effects of Priority Access in 802.11-Based Vehicular Ad-Hoc Networks, in Proc. of ACM VANET, 2004.
49. Haas, Z., Halpern, J.Y., and Li, L., Gossip-based Ad Hoc Routing, in Proc. of IEEE INFOCOM, 2002.
50. Bachir, A. and Benslimane, A., A Multicast Protocol in Ad Hoc Networks Inter-Vehicle Geocast, in Proc. of IEEE VTC, 2003.
51. Nadeem, T., Shankar, P., and Iftode, L., A Comparative Study of Data Dissemination Models for VANETs, in Proc. of ACM MOBIQUITOUS, 2006.
52. Chen, Z., Kung, H., and Vlah, D., Ad Hoc Relay Wireless Networks Over Moving Vehicles on Highways, in Proc. of ACM MOBIHOC, 2001.
53. Niculescu, D. and Nath, B., Trajectory Based Forwarding and Its Applications, in Proc. of ACM MOBICOM, 2003.
54. Wu, H., Fujimoto, R., Guensler, R., and Hunter, M., MDDV: A Mobility-Centric Data Dissemination Algorithm for Vehicular Networks, in Proc. of ACM VANET, 2004.
55. Little, T.D.C. and Agarwal, A., An information Propagation Scheme for VANETs, in Proc. of IEEE Intelligent Transportation Systems, 2005.