ROUTING IN WIRELESS NETWORKS WITH INTERMITTENT CONNECTIVITY

I. Cardei, C. Liu and J. Wu
Florida Atlantic University, Boca Raton, Florida, U.S.A.

Keywords:
delay-tolerant networks; challenged networks; routing protocols; network connectivity; transmission scheduling

Definition: *Wireless networks with intermittent connectivity (also called Delay or Disruption Tolerant Networks), are characterized by sporadic availability of end-to-end paths between end hosts.*

The Internet has been exceedingly successful in establishing a global communication network built on the concept of a common set of TCP/IP protocols. Within the last ten years there have been tremendous research efforts spent adapting the TCP/IP protocol stack to various types of wireless and mobile networks. Routing has been recognized as the most challenging problem in networks with a dynamic topology. Protocols, such as AODV [15], DSR [10], OLSR [4] and many others have been thoroughly analyzed in multiple scenarios. Their main limitation comes from the fact that, by design, they work only if there is a contemporaneous end-to-end path between endpoints. These protocols are able to find a route only if the destination router can complete the route discovery protocol (for on-demand routing protocols) or successfully disseminate link state advertisements (for table-driven routing). Node mobility or sporadic channel availability increases route instability, causing an increase in routing overhead and a reduction in end-to-end connectivity. Under these circumstances, routing protocols cannot keep up with the topology changes, and the overall network performance is reduced.

In this chapter we present an introduction to routing in networks with intermittent connectivity, and we cover several representative routing mechanisms. We begin by describing the main approach for message delivery in case of intermittent connectivity. After a brief overview of the Delay Tolerant Networking architecture, we describe the main classes of routing protocols for networking with intermittent connectivity and several representative solutions. *Deterministic routing* uses accurate estimates of time intervals when node links, called contacts, are available to schedule transmissions. *Stochastic routing* techniques are either *zero-knowledge*, where nothing is known about node contacts and state, or use *delivery estimation* to approximate a metric for end-to-end message delivery that contributes to more intelligent forwarding decisions. Active stochastic routing techniques rely on controlling the trajectory for some mobile nodes to pickup, carry and deliver messages to improve communication capability in sparse networks. For a comprehensive survey of the state-of-the-art in routing for DTN readers should consult [23].

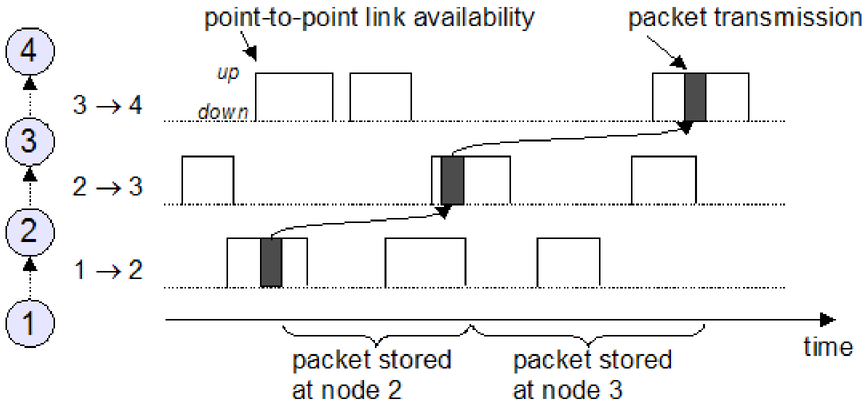

Figure 1. Packet delivery from source 1 to destination 4 is possible without a contemporaneous end-to-end path, provided nodes buffer packets until a link to the next hop becomes available.

A key reason why end-to-end communication is difficult in networks with dynamic topology is that network-layer IP packet delivery works only for as long as the end-to-end path is available. An IP packet will be dropped when it arrives at an intermediary node where, currently, no link to the next hop exists. This protocol design element restricts end-to-end delivery for a range of scenarios where packets can be buffered while in transit to the destination. Figure 1 illustrates such a scenario.

A route from source node 1 to destination 4 passes through nodes 2 and 3. Since there is no time instant when all three links are functioning, the standard IP forwarding approach would cause the packet to be dropped after reception at node 2. With a different approach, the packet could be buffered at node 2 until the 2 → 3 link becomes available. Similarly, the packet would wait at node 3 before it could be forwarded to the destination 4. Thus, even though no instantaneous source-destination path is ever available for the time considered, a simple *store-and-forward* protocol could deliver packets successfully. If the node is meanwhile moving, this approach is called *store-and-carry*.

Scenarios where network partition is frequent and sporadic connectivity is the norm are very common for a wide range of challenged networks, operating in difficult or "exotic" environments. Initial interest in developing effective communication protocols for networks with intermittent connectivity has been shown by NASA and DARPA in funding the Interplanetary Internet Project [1]. Their goal is to "define the architecture and protocols necessary to permit interoperation of the Internet resident on Earth with other remotely located internets residing on other planets or spacecraft in transit." An interplanetary internet encompasses ground fiber and satellite networks, earth – space links, and remote space-based wireless networks. It exhibits a wide range of link delays from the huge propagation distances, as well as sporadic link connectivity from planetary line-of-sight occlusion caused by the orbital motion of space-based communication assets.

Several other challenged networks with similar intermittent connectivity have attracted researchers' attention, such as MANETs, wireless sensor networks, acoustic underwater networks and networks for internet access in undeveloped areas. Wireless ad-hoc networks for disaster recovery also suffer from sporadic connectivity, caused mainly by node mobility and by a communication channel with variable quality. A common name for such networks with intermittent connectivity is Delay or Disruption Tolerant Networks (DTN).

Delay and Disruption Tolerant Network Architecture

The Delay Tolerant Networking Research Group [2] has been created as part of the Internet Research Task Force (IRTF) to address the architectural and protocol design principles needed for interconnecting networks operating in environments where continuous end-to-end connectivity is sporadic. Members of the DTNRG were instrumental in defining the initial DTN architecture. Kevin Fall was among the first to describe in [6] the main challenges facing current IP-based networks. 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, employ persistent storage to buffer bundles whenever a proper contact is not available for forwarding. The bundle layer is responsible for implementing reliable delivery and optional end-to-end acknowledgment. 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 [6] and the Internet Draft by Vint Cerf et al. [5].

Since 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 points out in [6] that routes in a DTN consist of a sequence of time-dependent communication opportunities, called contacts, during which messages are transferred from source towards the destination. Contacts are described by capacity, direction, the two endpoints, and temporal properties such as begin/end time, and latency. Routing in this 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 [5] and [6]. Persistent contacts are those always available. On-Demand contacts require some action in order to instantiate. 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. A scheduled contact is an agreement to establish a contact at a particular time, for a particular duration. Predicted contacts are based on a history of previously observed contacts or some other information.

As DTN routing must operate on a time-varying multigraph, message forwarding requires scheduling in addition to next-hop selection. To optimize the network performance, such as delivery rate or latency, DTN routing must select the right 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. The forwarding decision is more effective when it has better information on the current state of the topology and on its future evolution. 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 scheduled such that network performance is optimal and resource utilization is reduced by using unicast forwarding. At the other end of the spectrum, nodes know very little or nothing about the future evolution of the topology, and node movement is random or unknown. 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 absence of accurate current and future network state.

The next section describes the principles of operation of representative deterministic and stochastic DTN routing mechanisms.

Deterministic Routing Techniques

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. 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 protocols use single-copy unicast for messages in transit and provide good performance with less resource usage than stochastic routing techniques. 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. A major problem facing deterministic routing protocols remains the distribution of network state and mobility information under sporadic connectivity, long delays, and sparse resources.

Jain et al. present in [9] a deterministic routing framework that takes advantage of increasing levels of information on topology and traffic demand 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 there may exist multiple directed links between two nodes. Figure 2.a) illustrates a DTN scenario with 3 nodes connected by slow dialup links, infrequent and fast satellite links, and a bundle courier riding public-transportation, who is capable of delivering a large number of messages with latency measured in hours. The DTN multigraph is shown in Figure 2.b).

Figure 2.c) illustrates the time varying edge capacity $c\left(e_{13}^{sat},t\right)$ and $c\left(e_{13}^{dialup},t\right)$ for the four directed edges connecting node 1 to node 3.

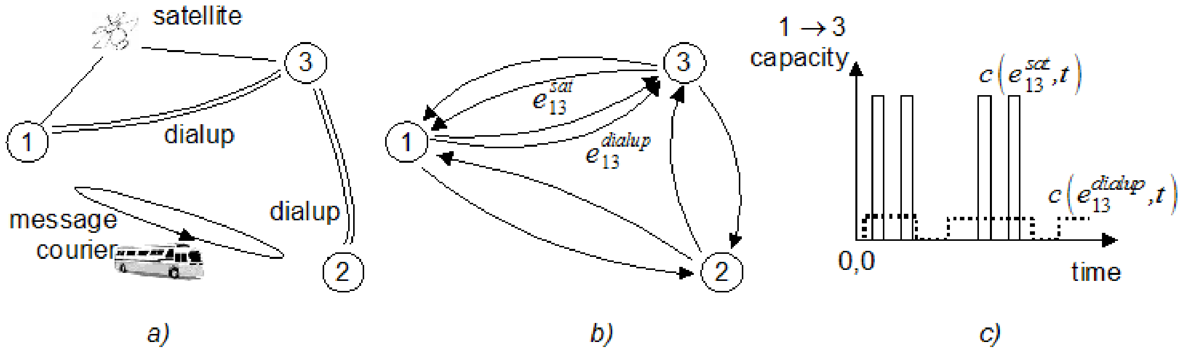

Figure 2. a) DTN topology with time-varying link capacity. Between nodes 1 and 3 the LEO satellite link provides higher throughput several times per day for only 10' at a time. The dialup lines are up all during low-fare times. A courier rides the bus and carries message bundles on a high capacity flash memory between nodes 1 and 2. b) the resulting DTN multigraph, with multiple directional edges between nodes 1 and 3. c) comparison of the time-varying capacity for directional edges 1 → 3.

The routing objective 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 the Dijkstra shortest path algorithm to support time-varying edge weights defined by the oracles available and propose six algorithms for finding the optimal contact path.

The first two algorithms from [9] assume time-invariant edge weights. 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 congestions and does not recompute routes for messages in transit.

The next four proposed partial-knowledge algorithms work with a time-varying edge cost, defined as the sum of the waiting, transmission, and propagation delays. The waiting delay includes the time waiting for a contact and the queuing delay. The cost for an edge *e* at time *t* is defined for a message of size *m* and for a root node *s:* $w'(e,t,m,s)=t'(e,t,m,s)-t+d(e,t)$, where *d* is the propagation delay, and *t*' is the earliest time the message for which the route is computed completes transmission. *t*' is the earliest time

the accumulated contact volume $\int_{x=t}^{t''} c(e,t)dt$ exceeds the total queued data. *t*' includes queuing delay and the time waiting for the corresponding contact :

t \llline ~ Int cSub { size 8{ ital x=t} } cSup { size 8{ ital t''} } {c left (ital e,t right) ital dx} >= left (ital m+Q left (ital e,t,s

$t'(e,t,m,s)=min¿$

¿

The edge capacity function *c*(*e*,*t*) and the propagation delay *d* are predicted by the Contact Oracle. The parameter *Q*(*e*,*t*,*s*) is the queue size for edge *e* predicted by node *s* at time *t*. The Earliest Delivery (ED) algorithm uses only the Contact Oracle and ignores queue occupancy: *Q(e,t,s)* ≡ 0. ED is prone to message loss due to buffer constraints, as the route for a message is not recalculated while the message is in transit.

The Earliest Delivery with Local Queuing algorithm (EDLQ) defines *Q(e,t,s)* to be equal to the local queue size at node *s,* 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 Earliest Delivery with All Queues (EDAQ) algorithm uses the Contacts and the Queuing oracles. *Q(e,t,s)* predicts the correct queue space for all edges at all times. In EDAQ, routes are not recomputed for messages in transit since the initial route predicts accurately 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 global and accurate distribution of queuing state. Limited connectivity also severely limits practical implementations of edge capacity reservations.

Simulations results indicate, as expected, that algorithms that use the knowledge oracles (ED, EDLQ, 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) performs close to the EDAQ algorithm. This means that similar network performance can be achieved without expensive queue state dissemination and capacity reservations.

In practice, contacts may be deterministically predictable only for a finite time horizon in the future, as trajectories and mission objectives may later change. Merugu et al. propose in [14] a deterministic routing framework where a space-time graph is built from predicted contact information. It starts with a time-varying link function $L_{ij}(t)$ defined as 1 when the link between two nodes is available and 0 otherwise. This function is defined for time t in a time interval [0, T], where T is the time horizon and the time is discretized in units τ. The space-time graph is build in $\lfloor T/\tau \rfloor$ layers, where the network nodes are replicated at each layer for each each time unit τ. Each layer has a copy of each network node. A column of these vertices maps to a single network node. Figure 3 illustrates the construction of the space-time graph for a simple two-node network where link A→B is available at times 1τ and 3τ for the duration of one time unit. 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. In Figure 3, the message delay is equal to the time unit.

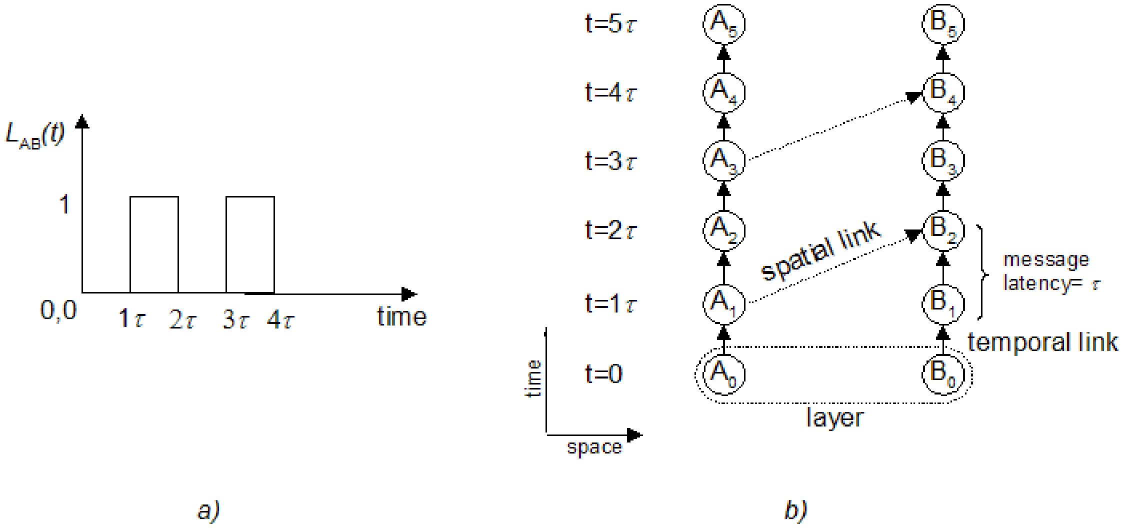

Figure 3. Construction of the time-space graph. a) a 2-node topology with time-varying link function $L_{AB}(t)$. b) the resulting time-space graph organized in node layers (one per time unit), temporal links – connecting the same DTN node across consecutive layers (buffering) – and spatial links, representing message forwarding between two different nodes.

Least cost routing in this DTN becomes equivalent to finding the least 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-Warshall 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 T, the size of the time horizon. This space-time routing approach is similar to the Earliest Delivery partial-knowledge algorithm from Jain et al. [9] 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.

Gnawali et al. propose in [7] 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 trajectories of satellites and other space assets. Link state updates with predicted contacts and their link performance 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. To support proximity routing for space assets in close formation, the authors propose a data-centric approach similar to directed diffusion [8].

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 since nodes may use outdated topology information. Cycles are avoided with path vectors. Deterministic DTN routing protocols are effective for a limited range of applications, where the contact schedule can be accurately modeled and predicted. Otherwise it is necessary to frequently disseminate nodes' state 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 of past experience.

Passive Stochastic Routing Techniques

Stochastic routing techniques can be passive or active, depending on whether node mission is changed in order to support message relay. *Passive routing* techniques do not interfere with node mission and do not change the node trajectory to adapt to traffic demands. Passive routing techniques rely in general 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 nothing or very little is known about mobility patterns.

We present first two passive stochastic routing protocols, *Epidemic Routing* and *Spray and Wait*, that do not need any information on 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.

Vahdat and Becker propose in [19] the *Epidemic Routing* protocol for message delivery in a mostly disconnected network with mobile nodes. Epidemic routing implements flooding in a DTN and got its name from 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 needs to pull from the other node. 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 contention for buffer space and transmission time. An approach to mitigate buffer space contention is for nodes to reserve a fraction of their storage for locally originated messages. 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.

Epidemic routing uses node movement to spread messages during contacts. With large buffers, long contacts or a low network load, epidemic routing is very effective and provides minimum delay and high success rate, 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. Tuning message TTL and buffer allocation allows an epidemic routing implementation to trade off message latency and delivery ratio. 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.

An approach to reduce the wasteful flooding of redundant messages in a DTN is presented by Spyropoulos et al. in [18]. A multi-copy, zero-knowledge routing protocol called Spray-and-Wait is introduced. Similarly to epidemic routing, this protocol forwards message copies to nodes met randomly during contacts in a mobile network. The main difference to 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 to a total of *L* distinct relays. In the wait phase all *L* nodes storing a copy of the message perform *direct transmission*. Direct Transmission [17] is a single-copy routing technique in DTN 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 the source will ever have contact with the destination node.

Spray and Wait initially spreads *L* copies of a message in epidemic fashion in order to increase the probability that at least one relay node would have a 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.

The optimal forwarding policy when nodes move randomly with identical and independent probability distribution (i.i.d.) is called *Binary Spray and Wait*. With this approach, the source node begins with L copies for each message. When a source or relay node *A* with n > 1 copies has contact with another node *B* that has no copies, *A* will hand over to *B* $\lfloor n/2 \rfloor$ copies and will hold on to $\lceil n/2 \rceil$ copies. When n = 1 a node will revert to direct transmission, meaning it will wait for a direct contact with the destination node. 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. Edges in the tree are formed by the transfer contacts. The more that nodes have multiple copies to distribute, the less the expected end-to-end delay will be. With the number of tree nodes fixed to $2^{1+\log L}-1$, a balanced binary tree has the maximum number of internal nodes and also the maximum number of nodes at every level. Therefore 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. At higher loads it performs much better than epidemic routing, since 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, energy are wasted on end-to-end message delivery. The drawbacks are 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 [13]. PROPHET works on the realistic premise that node mobility is not truly random. The authors assume 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, *P(A, B)*, 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*.

At the beginning of a contact, the two nodes (*A* and *B*) exchange the summary vectors (like in epidemic routing) and also the delivery predictability vectors. 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:

$$P(A,B)=P(A,B)_{old}+\left(1-P(A,B)_{old}\right)P_{init} .$$

P_{init} is a initialization constant between 0 and 1. For nodes that have not met for a longer time, their delivery probability should be reduced. The delivery probability is therefore aged with exponential decay: $P(A,B)=P(A,B)_{old}\gamma^{k}$ where $\gamma\in(0,1)$ is the aging constant and *k* is the length of the time interval since the previous aging. In addition, the delivery predictability has a transitive property that encodes the assumption that if nodes *A* and *B* have frequent contacts and nodes *B* and *C* have frequent contacts, then node *A* has a good choice to forward messages intended for node *C*. After exchanging delivery predictability vectors at the beginning of a contact, nodes *A* and *B* update their values for each other node *C,* using $\beta\in[0,1]$, a scaling constant that controls the impact of transitivity on delivery probability:

$$P(A,C)=P(A,C)_{old}+\beta\left(1-P(A,C)_{old}\right)P(A,B)P(B,C) .$$

When 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 FIFO policy for dropping old messages when new messages are received.

Transitive reinforcement of delivery probabilities based on prior contacts make this protocol perform better in simulations than epidemic routing, since 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 [3] and Zebranet [11].

A novel approach for delivery estimation is the use of a virtual Euclidean mobility pattern space, called *MobySpace,* proposed by Leguay et al [12]. 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 [16].

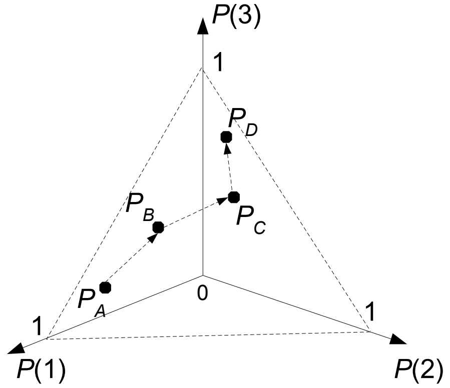

Figure 4. Routing in MobySpace (3D for locations 1, 2 and 3) from node A to destination node D. MobyPoints P_A ... P_D are situated in the plane P(1)+P(2)+P(3)=1. The message follows a contact route with points successively closer to P_D.

Citing studies of 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 more often in far fewer locations. When d→1 nodes have similar preference for all locations. The mobility pattern space has a dimension for each possible location, and the coordinate value 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.

Two points in MobySpace that have a small distance between them are more likely to have a contact than two nodes 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. Thus, the message takes a path through the MobySpace that brings it 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.

An example for routing using the mobility pattern space is shown in Figure 4. The network has three reference locations 1,2,3, visited by at least four nodes *A, B, C, D*. Each node knows the MobySpace coordinates for all nodes. Node *A* has a message to send to node *D*. When it encounters node *B* it forwards the message since $d(A,D)>d(B,D)$. Through successive contacts, the message eventually arrives at node *D*. Note that all points in a MobySpace with N dimensions are located on an (N – 1) dimensional hyperplane defined by $\sum P(i)=1$.

The MobySpace approach is effective only 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. While 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. Mobility pattern similarity does not guarantee 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 are effective dissemination of location probabilities for all nodes in a constrained DTN and high convergence time.

Active Stochastic Routing Techniques

Active routing techniques rely on controlling the trajectory of some nodes 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. In most cases the objective 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 . Similarly in wireless sensor networks, mobile nodes can traverse the sensing area and pickup/deliver measurements, queries and event messages. We review in the remainder of this chapter two DTN routing mechanisms that employ active node trajectory control.

Burns et al. introduce in [3] 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 nullspace or subsumption. Each controller adjusts the node trajectory such that its own objective is maximized.

- The Total Bandwidth Controller (φ_T) 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 (φ_U) selects a node that has the largest number of new

messages not yet forwarded to any other nodes.

- The Delivery Latency Controller (φ_D) picks the node with the highest average delivery time.
- The Peer Latency Controller (φ_P) selects the node least recently visited by an agent s.t. the traveling time to visit this node does not increase overall the peer latency metric.

The four controllers can be composed to optimize agent mission across performance metrics. To do that, controllers are first ordered according to their importance. With the *nullspace* 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 optimal 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 timestamp and flooded during contacts. Simulation results have shown that this approach is sufficient for low bandwidth and latency estimation error, but not enough to estimate correctly “last visit” times and location information. MV routing could be further improved with additional off-line or out-of-band network states. Another limitation of this approach is the key assumption that contact bandwidth is unlimited.

Zhao et al. describe in [21] 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 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 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. The same authors extend in [22] 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 [20].

Conclusions

This chapter presented an overview of some of the challenges facing routing in networks with intermittent connectivity, and described several routing solutions that used deterministic contact estimation, passive stochastic and active stochastic techniques. DTNs are a new area of wireless ad-hoc networking that shows great potential in many important applications. Routing and end-to-end message delivery in DTNs is possibly the most difficult problem in an environment where network resources are very limited and connectivity is scarce. The connectivity limitation affects the ability of distributing network-wide node and link information that could be used to optimize network operations. When contacts cannot be deterministically predicted, routing algorithms must rely on probabilistic methods that estimate future contacts with limited accuracy and on multi-copy forwarding that further strains the reduced network resources. In sparse DTNs, tasking dedicated nodes with ferrying messages improves the overall network capacity and reduces the delay.

Future research in DTN routing may address several remaining problems, such as effective integration of DTN techniques mostly-connected MANET routing protocols; Quality of Service and policy-based routing; statistical QoS guarantees and routing with probabilistic contact information.

Links

[1] The Interplanetary Internet Project, http://www.ipnsig.org/
[2] The Delay-Tolerant Networking Research Group, http://www.dtnrg.org

References

[3] B. Burns, O. Brock, and B.N. Levine. "Mv routing and capacity building in disruption tolerant networks." In IEEE INFOCOM 2005, volume 1, pages 398–408 vol. 1, 2005.

[4] T. Clausen and P. Jacquet. "Optimized link state routing protocol (OLSR)", 2003. RFC 3626, IETF Network Working Group.

[5] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, "Delay tolerant network architecture." draft-irtf-dtnrg-arch-05.txt, March 2006.

[6] K. Fall. "A delay-tolerant network architecture for challenged internets." In SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications, pages 27–34, New York, NY, USA, 2003. ACM Press.

[7] O. Gnawali, M. Polyakov, P. Bose, and R. Govindan. "Data centric, position-based routing in space networks." In 26th IEEE Aerospace Conference, 2005.

[8] C. Intanagonwiwat, R. Govindan and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," In Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (MobiCOM '00), August 2000, Boston, Massachussetts

[9] S. Jain, K. Fall, and R. Patra. "Routing in a delay tolerant network." In SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications, volume 34, pages 145–158. ACM Press, October 2004.

[10] D. B Johnson and D. A Maltz. "Dynamic source routing in ad hoc wireless networks." In Mobile Computing, volume 353. Kluwer Academic Publishers, 1996.

[11] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein. "Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet." In The 10th international

conference on Architectural support for programming languages and operating systems (ASPLOS), pages 96–107. ACM Press, October 2002.

[12] J. Leguay, T. Friedman, and V. Conan. “Evaluating mobility pattern space routing for DTNs”. In INFOCOM 2006

[13] A. Lindgren, A. Doria, and O. Schelén. “Probabilistic routing in intermittently connected networks.” In SIGMOBILE: Mobile Computing Communications Review, volume 7, pages 19–20. ACM Press, July 2003.

[14] S. Merugu, M. Ammar, and E. Zegura. “Routing in space and time in networks with predictable mobility.” Technical report, Georgia Institute of Technology, 2004. GIT-CC-04-7.

[15] C. Perkins, “Ad hoc on demand distance vector (AODV) routing”, 1997. Internet Draft.

[16] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. “A scalable content addressable network.” In Proc. SIGCOMM, 2001.

[17] R. C. Shah, S. Roy, S. Jain, and W. Brunette. “Data mules: Modeling a three-tier architecture for sparse sensor networks.” In IEEE International Workshop on Sensor Network Protocols and Applications, pages 30–41, May 2003.

[18] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. “Spray and wait: an efficient routing scheme for intermittently connected mobile networks. “In the ACM SIGCOMM workshop on Delay-tolerant networking, pages 252–259, New York, NY, USA, 2005. ACM Press.

[19] A. Vahdat and D. Becker. “Epidemic routing for partially connected ad hoc networks.” Technical report, Duke University, Apr. 2000. CS-200006.

[20] J. Yang, Y. Chen, M. Ammar, and C. Lee. “Ferry replacement protocols in sparse MANET message ferrying systems.” In IEEE WCNC, 2005.

[21] W. Zhao, M. Ammar, and E. Zegura. “A message ferrying approach for data delivery in sparse mobile ad hoc networks.” In The 5th ACM international symposium on Mobile ad-hoc networking and computing (MOBIHOC), pages 187–198, New York, NY, USA, 2004. ACM Press.

[22] W. Zhao, M. Ammar, and E. Zegura. “Controlling the mobility of multiple data transport ferries in a delay-tolerant network.” In INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE, volume 2, pages 1407–1418 vol. 2, 2005.

[23] Z. Zhang, “Routing In Intermittently Connected Mobile Ad Hoc Networks and Delay Tolerant Networks: Overview and Challenges”, IEEE Communications Surveys, vol. 8, 2006