Routing Schemes for Delay-Tolerant Networks - An Applications Perspective

Technical Report 2012-588

Ph.D. Depth Paper

Ashraf E. Al-Fagih
Hossam S. Hassanein

Telecommunications Research Lab (TRL)
School of Computing
Queen's University
Kingston, Ontario, Canada, K7L 3N6

Submitted: December 2010

Abstract

Delay Tolerant Networks (DTNs) represent a class of wireless networks that experience frequent and long lasting partitions due to sparse distribution of nodes in the topology. A traditional TCP/IP setting assumes the definite existence of a contemporaneous end-to-end path between any source-destination pair in the network. Any setting that violates this assumption may be considered as a potential application for the DTN architecture. To cope with this situation, DTN nodes utilize a store-carry-forward approach in which messages are buffered for extended intervals of time until an appropriate forwarding opportunity is recognized. Numerous studies have tackled the challenging problem of routing in DTNs. Routing proposals include stochastic approaches such as random, spray-and-wait and epidemic routing, or deterministic approach such as history-based, model-based, coding-based and variations of these approaches. The number of routing schemes in the literature is increasing rapidly without a clear mapping of which is more suitable for any of the vast array of potential DTN application. This document surveys the main routing schemes in the DTN literature. It provides a detailed insight to the DTN approach and describes in some depth the policies and strategies proposed to manage buffers and queues in DTN nodes. Then, the predominant DTN real-life applications are presented and discussed. The aim of this document is to create a classification basis for the most prominent DTN applications and to map major DTN routing schemes to these categories according to some influential network attributes and characteristics that are related particularly to the type of application in hand.

Contents

List of Figures

List of Tables

List of Acronyms

TCP/IP	Transmission Control Protocol / Internet Protocol
ICN	Intermittently Connected Network
DTNs	Delay Tolerant Network
WSNs	Wireless Sensor Network
EMN	Exotic Media Networks
IPN	Inter-Planetary Network
RF	Radio Frequency
UWB	Ultra Wide Band
GPS	Global Positioning System
DSL	Digital Subscriber Line
BDP	Bandwidth-Delay Product
RTT	Round Trip Time
SCF	Store-Carry-Forward
FIFO	First In First Out
MOFO	Evict Most Forwarded First
MOPR	Evict Most Favourably Forwarded First
SHLI	Evict Shortest Life Time First
LEPR	Evict Least Probable first
MRS	Most Recently Seen
LRU	Least Recently Used
MFS	Most Frequently Seen
LFU	Least Frequently Used
WSF	Weighted Storage and Frequency
MANETs	Mobile Ad Hoc Networks
OLSR	Optimized Link State Routing
AODV	Ad hoc On-Demand Distance Vector (Routing)
LAR	Location-Aided Routing
STAR	Source-Tree Adaptive Routing
PRoPHET	Probabilistic Routing Protocol using History of Encounters and Transitivity
MBR	Model Based Routing
MF	Message Ferry
NIMF	Node-Initiated Message Ferrying
FIMF	Ferry-Initiated Message Ferrying
VBR	Vector Based Routing
FVR	Flooding-based Vector Routing
HVR	History-based Vector Routing
DF	Delegation Forwarding
SWIM	Shared Wireless Info-station Model
AUV	Autonomous Underwater Vehicle
DT	Drop Tail
DO	Drop Oldest
DY	Drop Youngest
ND	N-Drop (ND)
BMP-MIPN	Buffer Management Policy for Mars Intelligent Proximity Networks
GBSD	Global knowledge Based Scheduling and Drop

1. Introduction

Delay (or disruption) tolerant networking, provides an alternative approach to a variety of emerging wireless applications and architectures that challenge the limitations facing the transport and routing layers in the TCP/IP model.

The traditional Internet model assumes low error rates, low propagation delays and, most importantly, a steady end-to-end connection between any source/destination pair of nodes. However, a class of *challenged networks* which violates one or more of these assumptions are becoming increasingly in demand and are being underserved by the TCP/IP paradigm [17]. Such networks mainly suffer from frequent temporary partitions and are referred to as Intermittently Connected Networks (ICNs). This phenomenon is particularly apparent in rural areas such as wild habitats and villages that lack basic infrastructures.

Delay tolerant networks (DTNs) represent a class of infrastructure-less wireless systems that support the functionality of networks experiencing frequent and long lasting partitions. DTNs are intended to deal with scenarios involving heterogeneity of standards, intermittent connectivity between adjacent nodes, lack of *contemporaneous* end-to-end links and exceptionally high delays and error-rates. Moreover, nodes operating in challenged environments are extremely limited in their resources; such as CPU processing power, memory and network capacity. A DTN setting has to account for all or some of these factors. [1] [22] [48].

DTN protocols and architectures [8] [43] are intended to achieve interoperability and eventual connectivity to a range of complex applications that include:

- Wireless sensor networks (WSNs) deployed in wildlife tracking or in extreme regions (e.g. volcanic and underwater areas).
- Mobile Ad-Hoc networks connecting remote and rural communities via GPSs, cellular devices and portable storages.
- Exotic Media Networks (EMNs) interconnecting extra-terrestrial nodes such as satellites and deep space probes in Inter-Planetary Networks (IPNs).

Wireless DTN technologies include radio frequency (RF) such as ultra-wide band UWB, free-space optical and acoustic (sonar or ultrasonic technologies) [43]. It is worth mentioning here that the IPN project [9] launched in 1998 by NASA's Jet Propulsion Laboratory may be considered as the first of a broader class of networks that also includes terrestrial scenarios and was originally termed then as DTNs. The philosophy of these networks was stated then as to use storage capacity, a variety of protocol techniques, replication and parallel forwarding and many other methods to overcome communication impairments. [16]

Each of the potential field of applications mentioned above is intended to operate under stressful circumstances and in environments that are considered to be challenging for ordinary wireless nodes within a traditional network settings.

DTN architecture represents an attempt to extend the reach of networks. It promises to enable communication between instances of such challenged networks and to act as an integral platform between instances that originally adopt heterogeneous or inconsistent standards, even if they exist in territories lacking a proper communication infrastructure. The main purpose of the DTN approach is to provide a means for message delivery in such challenged settings.

1.1 Characteristics of Challenged Networks

Challenged networks may be characterized by any combination of the following [43] [13]:

- Intermittent connectivity: If there is no consistent end-to-end path between the source and destination -a phenomenon known as *network partitioning*-, end-to-end communication using the TCP/IP protocols does not work. Other protocols are required.
- Asymmetric data rates: The Internet does support some forms of asymmetric bi-directional data, as in cable TV or asymmetric DSL access. But if asymmetries increase then they will hinder traditional interactive protocols such as TCP.
- High error rates: If bit errors occur on links, then they either require correction. Hence, consuming more bandwidth, or even retransmitting the entire packet, resulting in more network traffic.
- Ambiguous mobility patterns: Unlike the case with public bus services that maintain fixed routes or planetary trajectories, future behaviour of a node is not fully known for many DTN applications. It is widely assumed, however, that node mobility patterns (while random) are generally recurrent.
- Long or variable delay: Long propagation delays between nodes, in addition to variable queueing delays at node buffers, all create end-to-end path delays that far exceed the threshold levels usually tolerated by Internet protocols and applications that rely on quick return of acknowledgements.

Delay has been purposely left to the end of the list above for further elaboration. Intermittent connectivity is one of the major DTN characteristics and has a highly passive effect on its performance [24]. DTN research has been mainly motivated -as the name suggests- by this vital performance metric.

A handshake procedure is required to establish, and then terminate, a TCP connection between a client/server pair. Data transfers do not start until after a complete round-trip following the ACK from the client to the server. In the case of typical Internet, acceptable delays are in the order of milliseconds. However, as the delay approaches a specific TCP time-out value t_0 (implementation-dependent), then establishing a connection becomes impossible.

Moreover, if the connection is established, and the client disappears during the connection for any given reason, then the source will release all the provisioned resources for this particular connection session.

The Bandwidth-Delay Product (BDP) may be used to measure a protocol's efficiency in utilizing available bandwidth to transmit data. It represents the product of the Round Trip Time (RTT) and the bottleneck link bandwidth:

$$\textbf{\textit{BDP = Average Packet Transmission Time (Sec)}} \times \textbf{\textit{Connection Bandwidth (Bit/Sec)}} \qquad (1)$$

It can be seen from Equation 1 that as delay increases, the TCP protocol wastes more and more bandwidth during large round-trip times. We may imagine how worse the situation would be if TCP was put in charge for an ICN scenario, where links are highly subject to intermittent connectivity and hence, the network becomes frequently partitioned. There would be an extremely high probability of a series of unsuccessful handshakes, to say the least.

In this report, delay is defined as the end-to-end latency of data transmission contributed by three major factors [20]:

- Unavoidable intrinsic characteristics of the transmission medium, such as *light-trip time*. This is the elapsed time it takes for a signal to travel from Earth to a spacecraft (or other body) and back to the starting point.
- System's geometry (e.g. long distances).
- Temporary storage of packets in nodes' buffers while on their way from source to destination.

"Disruption" has been alternatively used instead of "delay" to represent the letter "D" in DTNs. *Disruptions* are defined as sudden failures of operating connections or denials of connections caused by brief variations of the system's characteristics such as power failures, positioning inaccuracies, transient obstructions, network topology changes, nodal destructions or nodal mobility. The next section is dedicated to this last characteristic, in particular.

1.2 The Issues of Mobility and Storage

In many DTN applications, nodes are able to move across the given plane either randomly or according to a deterministic manner. This is a double-edged feature. On the one hand, mobility helps achieving the ultimate goal of message delivery by bringing nodes closer to one another and facilitating their ability to relay/receive data packets.
In DTN context, data packets are referred to as *bundles*. A bundle contains all necessary information required for routing and delivery. In this concept, a complete data entity is bundled into one potentially large message. This is to ensure not splitting semantically related data into separate small-sized packets because transfer delays may differ and in case of the loss of one packet the already transferred data can't be used either. [57]
Figure 1 shows how the mobility of nodes can be used to eventually deliver a message to its destination. In this figure, node *A* has a message (indicated by the node being shaded) to be delivered to node *D*, but no path exists between nodes *A* and *D*. As shown in subfigures (a-d), the mobility of the nodes allow the message to first be transferred to node *B*, then to node *C*, and finally node *C* moves within the range of node *D* and can deliver the message to its final destination.
On the other hand, mobility causes constant topology shifts and different links come up and down over time between adjacent nodes. Not to mention the energy constraints imposed by continuous mobility. Therefore, bundles may have to be buffered for extensive periods of time by intermediate nodes, contributing to the delay discussed above. The mobility of these intermediate nodes must be exploited via exchanging data packets between the nodes as they meet to bring them closer to their destination.

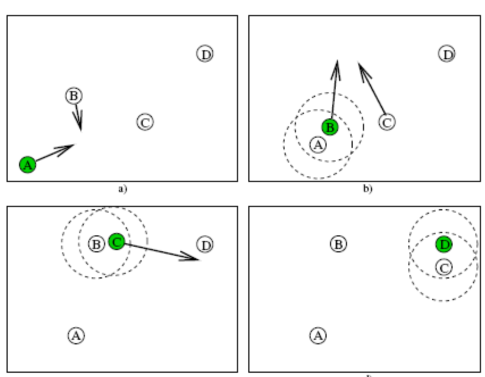

Figure 1 - Transitive nodal communication (from [32])

To achieve *eventual delivery* and cope with intermittent connectivity constraints, the traditional store-and-forward routing approach is extended under the umbrella of DTNs to *store-carry - forward (SCF)* [46]. A next hop may not be immediately available for the current node to forward the data. Thus, the node will need to buffer the data until the next link is established and the courier node gets an opportunity to forward the data.

The DTN architecture [8] [43] implements message switching by adding a new layer called the *bundle layer* on top of the region-specific lower IP layers. For a detailed description of the DTN bundle layer the reader can refer to [8].

In general, DTN routers require constant storage due to any of the following [43]:

- Long-term absence of a communication link for the next hop.
- Lack of symmetry. A node in a communication pair may send or receive data much faster than or more reliable than the other node.
- Need to re-transmission as a result of delivery errors or failure at recipient's side.

Buffering is expected to be maintained for extended durations of time that range from several hours to even days or months, depending on the application at hands. Yet, buffer overload is a severely panelizing problem as it exponentially increases the packet drop rate. Hence, buffer management schemes in DTNs had their share of research as will be presented in the section following.

Moreover, understanding node mobility characteristics will lead to a better understanding of how to optimize performance and routing in DTNs [14]. In the general context of mobile wireless networks, several mobility models have been proposed. Those models attempt to mimic the movement of mobile entities either individually or as a group. The area of mobility modeling is, however, out of the scope of this report.

1.3 Routing or Forwarding?

In the context of DTNs, the concepts of *routing* and *forwarding* are more intertwined and inter-related. Traditionally and also technically, forwarding is a single router's local action to find the next hop within the set of intermediate nodes, while routing is a network-wide process involving finding an end-to-end path between the source-destination pair [29].

From the perspective of DTNs, questioning the very suitability of using the term "routing" is repeatedly raised [47]. Routing usually involves tasks such as packet delivery, adaptation to topology changes, avoiding loops and congested links and minimizing routing overhead. Given the extreme limitations of DTN settings, none of these routing-related tasks are achievable. Moreover, given the absence of instantaneous end-to-end connectivity between any pair of nodes, there is simply no way to know if a sent-out data bundle will reach its intended destination or if the currently available forwarding opportunity is the best. Therefore, routing in DTNs may be referred to as rather an opportunistic forwarding algorithm based on a set of next hop selection rules aiming at delivering a bundle to its initiated destination [31].

The main assumption here is that sending a bundle forward should increases, the likelihood of its delivery. Same situation is highly possible to occur at the bundle's second hop and further next hops until it is ultimately delivered. The DTN forwarding problem requires deciding the best next-hop candidate among the neighbouring nodes, and deciding the most suitable time instant in

which a data bundle is to be forwarded to that next-hop. These decisions are related to the next hop's likelihood of successful delivery to the final destination, and its contribution in the overall network's performance optimization of a pre-defined metric.
Nevertheless, a bad forwarding decision may cause the bundle to be delayed indefinitely. Thus, buffer management is a fundamental part of this argument. Several DTN buffer management schemes have been proposed. These will be described as required in the following sections.

1.4 Paper Organization

The reminder of this report discusses routing and forwarding in DTNs including routing schemes, queueing policies and forwarding strategies. It then focuses on DTN implementation and lists some application examples before conducting a comparison between routing schemes and their most suitable DTN disciplines. Section 2 covers queueing and forwarding issues in DTNs. DTN forwarding solutions are presented by listing the main queueing policies and forwarding strategies utilized in DTN buffer management. In Section 3, the need for particularly designed DTN routing schemes is explained via comparing DTN requirements against traditional IP approaches. This includes an observation of the role of mobility knowledge in specifying the appropriate scheme. Then, major routing schemes are surveyed in detail. They are compared against each other in terms of some performance metrics such as delivery ratio, delay and nodal mobility patterns. Section 4 describes in more details the relation between queueing policies and routing schemes by surveying the policies most appropriate to each of the routing schemes mention in Section 4. Section 5 portrays main DTN fields of application. Several real-life scenarios are described and being traced in the DTN literature. In Section 6, we categorize and compare application scenarios. The application disciplines are compared against some networking characteristics to establish common factors between them. Once this categorization is done, a mapping between each application category and its most-suitable routing scheme is conducted. Finally, Section 7 concludes this report.

2. Queueing Policies and Forwarding Strategies

Over many DTN routing studies, it has been explicitly mentioned that choosing the correct buffer management policy and forwarding strategy, and combining them to the suitable routing method, represent the best recipe for improving performance in terms of message delivery, overhead and end-to-end delay [27] [31] [48].

As mentioned earlier in Section 1.2, store-carry-forward (SCF) is a natural approach that extends the traditional store-and-forward routing. In SCF, a next hop may not be instantly available for the current node to forward the data. The node will need to buffer the data until there is a forwarding opportunity. Hence, DTN protocols assume that each node maintains a buffering queue.

This buffering is characteristically proposed to cope with the DTNs' challenging constraints. For instance, typical small buffer sizes are obviously unsuitable under such constraints. Instead, to enhance data delivery, DTN nodes are augmented with permanent storage capabilities and equipped with relatively large buffer sizes enabling them to indefinitely hold data bundles until they can be further forwarded according to the SCF scheme.

Buffering, however, does not represent a permanent solution. Massages have to be rapidly and efficiently forwarded to achieve the optimal goal of delivery, primarily, and also to avoid buffer overflows. This later issue is of significant importance.

Due to the link-state-uncertainty situation discussed earlier, nodes are forced to distribute multiple message copies to other neighbouring nodes in an attempt to increase the bundle's delivery probability, a technique referred to as *flooding* (will be discussed in Section 3.3). Flooding the network haphazardly causes a rapid buffer overflow and therefore an increase in the drop rate per node. Some DTN routing approaches, however, assume unlimited buffer resources [41]. This assumption, of course, is neither applicable nor realistic. Buffer space is in fact a crucial DTN resource to be effectively managed.

Buffer management is responsible for deciding for each node which message to pop out of the queue. When to do so is mainly a forwarding agent's decision [31]. Buffer manager also decides, in case of congestion, which messages to drop from the queue.

This document isn't mainly intended to discuss queueing policies in DTNs. Nevertheless, buffer space is a critical resource that highly impacts the overall performance of DTNs. It is interesting to notice, though, that there has not been much emphasis on the development of efficient buffer management schemes and forwarding strategists compared to routing in the related literature. However, for the sake of providing a general overview on the issue, several DTN queueing policies and forwarding strategies are discussed in the following three sections.

2.1 Queueing Policies

Local queueing policies (i.e. policies on nodal level) can define rules to be followed either at one or both of the two following levels [22]:

- The forwarding level: If a node is congested, then bundles must not be forwarded to it temporarily. Some bundles may also be transferred from a congested node to other nearby nodes until congestion is resolved.
- The nodal buffer level: Here, the most appropriate bundles to be dropped are identified, including the ones being received, so as to reduce buffer space usage.

Reference [31] proposes the following queue management policies defining which message should be dropped if the buffer is full when a new message has to be accommodated:

- *FIFO - First in first out:* The message that was first entered into the queue is the first message to be dropped.
- *MOFO - Evict most forwarded first:* This policy requires keeping track of the number of times each message has been forwarded. The message that has been forwarded the most is the first to be dropped, thus giving messages that have not been forwarded fewer times a chance.
- *MOPR - Evict most favourably forwarded first:* This can be considered to be a weighted version of MOFO, where instead of increasing a counter by one each time a message is forwarded, it is increased by according to a *delivery predictability: P* the receiving node has for the message. Every node maintains a value *FP* (initialized to zero) for each message in its queue. Each time the message is forwarded, *FP* is updated according to the equation:

$$\boldsymbol{FP = FP_{old} + P} \qquad (2)$$

  The message with the highest *FP* value is the first to be dropped.
- *SHLI - Evict shortest life time first:* In the DTN architecture, each message has a timeout value which specifies when it is no longer useful and should be deleted. If this policy is used, the message with the shortest remaining life time is the first to be dropped.
- *LEPR - Evict least probable first:* Since the node is least likely to deliver a message for which it has a low *P*-value, drop the message for which the node has the lowest *P*-value.

The queue manager may be set in a manner that combines more than one queueing policy in an ordered set, where the first policy is used primarily; the latter policies are used –in order- only if there is a need to break a tie between messages with the same eviction priority assigned by the primary policy, and so on. As an example, one queueing policy could be {MOFO; SHLI; FIFO}.

2.2 Forwarding Strategies

Finite bandwidth and unexpected connection interruptions may not allow a node to transmit all the messages it would like to forward. In such cases, the order in which the messages are transmitted is important.

Forwarding strategies are mostly related to the *delivery predictability*. Considering a node *A* forwarding to a node *B* a bundle *M* that is ultimately destined to a node *D*. The delivery predictability *P(A,B)* is the expectation degree assessed by *A* with respect to *B*'s ability to deliver *M* to *D*.

Some forwarding strategies were also defined in [31] and [27]. Note that if the node being encountered is the destination of any of the messages being carried, those messages should be delivered to the destination irrespective of the forwarding strategy being used. Nodes do not delete messages after forwarding them as long as there is sufficient buffer space available (since it might encounter a better node, or even the final destination of the message in the future), unless the node to which a message was forwarded was its destination.

The following notations are used in the discussions below: *A* and *B* are the nodes that meet, and the strategies are described as followed by node *A*. The destination node is *D*. $P_{(X,\ Y)}$ denotes the delivery predictability that a node *X* has for a destination *Y*.

- ***GRTR**:* Forward the message only if $P_{(B,D)} > P_{(A,D)}$.When two nodes meet, a message is sent to the other node if the delivery predictability for the destination of the message is higher at the other node.

- ***GRTRSort**:* Select messages in descending order of the value of $P_{(B,D)} - P_{(A,D)}$. Forward the message only if $P_{(B, D)} > P_{(A,D)}$.
  This strategy is similar to GRTR, but it processes the messages in the message queue in a different way. While GRTR scans the queue in a linear way, starting by deciding whether or not to forward the first message, and the continuing like that through the queue, this strategy looks at the difference in *P*-values for each message between the two nodes, and forwards the messages with the largest difference first. This allows a node to transmit messages with most improvement in delivery predictability first.

- ***GRTRMax**:* Select messages in descending order of $P_{(B,D)}$. Forward the message only if $P_{(B,D)} > P_{(A,D)}$.
  This strategy begins by considering the messages for which the encountered node has the highest delivery predictability.
  The motivation for doing this is the same as in GRTRSort, but based on the idea that it is better to give messages to nodes with high absolute delivery predictabilities, instead of trying to maximize the improvement.

- ***COIN:*** Generate a variable $X \in U(0, 1)$ and forward the message only if $X > 0.5$.
  This strategy is similar to the Epidemic Routing (to be discussed in Section 2.4.1) but to reduce the number of transfers, there is a "coin toss" that determines if a message should be forwarded or not. This strategy does not consider the delivery predictabilities in making its decision.

The relation between the queueing and forwarding strategies and specific routing schemes will be discussed later in the conclusion of this document after reviewing these schemes (Section 3). For the remaining of this section, we draw an analogy between routing and buffering policies.
The authors in [21] describe several routing strategies for DTNs that are inspired by common cache replacement policies. Their proposed algorithms are meant to work by having each node maintain a list about neighbouring nodes that are encountered. The list is sorted according to a certain “cache replacement policy” as described below and is broadcast over the network. These caching policies (listed bellow) provide a ranking criterion whose inverse is taken as routing cost. The resulting values are used to assign weights to the available edges.

- ***Most Recently Seen (MRS)***

This algorithm is analogous to the standard Least Recently Used (LRU) cache replacement policy. The neighbour list is sorted by the time last seen in the normal LRU fashion. The weight for an edge e_{ij} between node *i* and node *j* at time *t* is given by:

$$w(e_{ij}, t) = t - lastSeen_{ij} \qquad (3)$$

In MRS, the neighbour lists represent the most recent encounters with neighbouring nodes. It should be expected that these snapshots will vary substantially over time, as those lists are being broadcasted and propagate through the network while the nodes keep on moving and changing the overall topology. Although a certain node currently would make a good next-hop to some destinations, it is likely that this is no longer the case once the neighbour list has propagated through the network and new routing tables have been computed. Furthermore, the edge weights assigned by this policy to possible single-hop links do not consider the delay between each hop in the route. Instead, routing based on this out-of-date information could potentially lead to very long worst-case delays and large buffer requirements to implement the store-and-forwarding.

- ***Most Frequently Seen (MFS)***

This algorithm is analogous to the Least Frequently Used (LFU) cache replacement policy. In MFS, each node maintains a *counter* for each neighbour node and the counter is incremented on each encounter. Thus, the list entries in node *i* are on the format $\{j, counter_{ij}\}$. The lists are sorted according to the *counter* values.

Since the *counter* value represents the frequency of encounters, its inverse will be a relative measure of the expected delay to the next encounter. Consequently, in an attempt to minimize the delay, the inverse of the counter values is used as hop cost in the construction of shortest-path routes. The edge weight between node *i* and node *j* at time *t* is given by

$$w\,(e_{ij}\,,\,t) = 1 \,/\, counter_{ij}\,(t) \qquad (4)$$

If node movements are recurrent, it is expected that the routes given by the MFS strategy converge to reflect the average periods of node encounters in the network. MFS is expected to give routing tables with better average delays and smaller storage requirements than MRS.

- *Weighted Storage and Frequency (WSF)*

In the MRS and MFS algorithms, the edge weights are defined as measures of the delay associated with routing a message between two nodes. However, for memory-constrained systems, the buffering needed to implement the store-and-forwarding in DTNs should also be considered in the routing.

This was achieved in [21] by defining modified edge weights as:

$$w(e_{ij}\,,\,t) = B_j \,/\, counter_{ij}\,(t) \qquad (5)$$

where B_j is the buffer size at node *j*. Using this strategy, routes will be chosen to minimize the weighted cost of storage and end-to-end delay. This Weighted Storage and Frequency (WSF) strategy is implemented by appending the current buffer size of each node to the packets that are broadcast to share neighbour information.

It should be noted the WSF strategy requires less storage than MFS, since the buffer size is explicitly considered in the routing metric. As a consequence, the average delay will be larger for WSF. However, the weighted routing cost (delay and buffer space) is expected to be smaller for WSF compared to MFS and MRS. Since the WSF routing tables depend on the buffer sizes they will be less stable than for the MFS strategy.

- ***Aging***

Finally, the routing strategies should be complemented with an aging factor to adapt to node mobility. The use of this strategy will reduce the effect of old node encounters on the current neighbour lists. Thus, allowing the routing tables to converge more quickly in response to dynamic changes.

3. Routing in DTNs

The routing problem in DTNs may appear as a standard problem of dynamic routing in mobile Ad Hoc networks (MANETs), but with extended link failure times. This is, however, not the case [22]. To accommodate the dynamic topology in MANETs, an abundance of routing-layer protocols such as OLSR [10], AODV [33], LAR [28], STAR [18], and many others, has been proposed. Some of those routing protocols adopt a reactive approach by not taking the initiative for finding a route to a destination until it is required. AODV and DSR fall under this category. Proactive protocols, on the other hand, use periodic exchange of control messages. They provide the required route to destinations immediately but at the cost of bandwidth consumed in periodic topology updates. Protocols such as OLSR, DSDV and STAR fall under this latter category.
For all these routing protocols, however, it is implicitly assumed that the network is connected and there is a contemporaneous end-to-end path between any source/destination pair. In the standard dynamic routing problem, the topology is assumed to be indefinitely connected (with the omission of very short intervals of partitioning), and the objective of the routing algorithm is to find the best currently available full path to move traffic from one end to the other.
Unfortunately, none of these assumptions stand in a DTN setup. DTN nodes mostly lack network state information (i.e. information about other nodes in the network, network topology... etc.). Routing protocols such as AODV and OLSR do not work properly in DTNs, since under these protocols, when packets arrive and no end-to-end paths for their destinations can be found immediately, these packets are simply dropped.
Figures 2 and 3 illustrate an AODV example that shows how this particular routing protocol does not fit an intermittently connected topology. The two figures assume a path *P* to be discovered on the demand of a source *S* to send data to destination *D*. If any node along *P* disappears for a particular reason, the protocol will totally fail. So to generalize in this context, a set of paths traversing nodes that may disappear from the network for any reason becomes invalid and remain so until those nodes get in touch again.

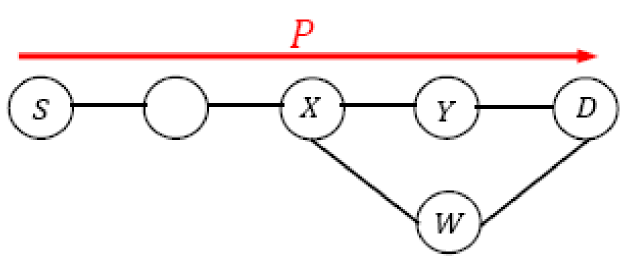

Figure 2 -Normal flow from S to D over path P (from [33]).

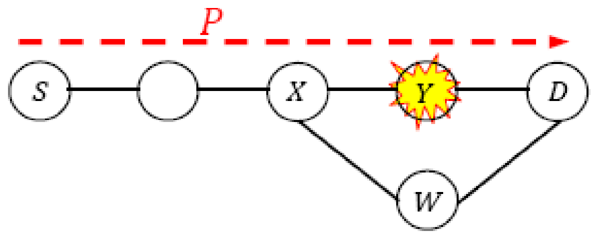

Figure 3 - Node Y fails and hence path P fails (from [33]).

In Figure 3, node *X* has another route to node *D* through node *W*. However, *X* cannot send its packets to *W* since the end-to-end AODV protocol determined earlier (Figure 2) that the next hop of the packets at *X* is *Y*. Hence, all of *X*'s packets are lost and the protocol fails.

From a DTN perspective, *Y* may be assumed as a node with no continuous power source and hence, has to go to *sleep mode* in order to save power. During the instance of time in which *Y* was still active, a protocol like AODV determines a path *P* from *S* to *D* through *Y* and stores it in *S*'s routing table. This will cause *S* to transmit packets over *P* that are never received by *D* and accordingly waste a lot of valuable resources. The problem gets more serious if several nodes of a MANET similar to *Y* are the only ones physically in the proximity of an Internet gateway and they all happen to be in sleep mode, then the entire MANET is disconnected.

The scenarios proposed above apply to Internet routing protocols other than AODV. In a DTN, an end-to- end path is only available intermittently and the act of routing/forwarding is performed over time to achieve *eventual delivery.* This is accomplished by employing long-term storage at the intermediate nodes in a store-carry-forward approach as mentioned earlier. Hence, point-to-point forwarding is an essential and integral part of any DTN routing scheme. And as routing in DTNs is discussed, it is equally necessary to elaborate on forwarding and buffer-management techniques in DTNs.

3.1 Types of DTN Routing

The DTN routing problem may be viewed as an optimization problem where edges may be unavailable for extended periods of time and a storage constraint exists at each node [22]. This turns DTN routing to a considerably different and more challenging problem. Consequently, substantial effort has been put to develop new routing protocols and system architectures for DTNs [8, 13, 14, 16, 17, 20, 23, 24, 25, 32, 41, 43, 48, 49].

As mentioned earlier, there is no way for a source DTN node to tell in advance if an end-to-end path exists to an intended destination node. DTN routing has been described as a set of opportunistic forwarding decisions and is performed via SCF by moving the message closer to the destination one hop at a time. As a result of such a situation, the knowledge of the mobility patterns of a set of nodes within a given DTN partition is of utmost importance in this regard. Mobility knowledge will be further discussed in the next section.

A number of classification schemes have been proposed DTN routing protocols. One common method of classification is according to the mobility behaviour being either *deterministic* or *stochastic* [48] If the mobility behaviour of the nodes is deterministic and known, or at least predictable, then message transmission (i.e., when and where to forward packets) can be scheduled ahead of time so that some optimal objective can be achieved. In a stochastic case, on the other hand, the future topology of the network (as a time-evolving graph) is assumed to be random or absolutely unpredictable. Thus, nodes are supposed to roam across the plane carrying the data in anticipation of a *suitable* forwarding chance.

As mention earlier in Section 1.3, the terms routing and forwarding are used interchangeably in the DTN jargon. Mostly, the work done on routing in DTNs can be divided into two broad categories: *Flooding* protocols and *forwarding* protocols [23]. Flooding (or random routing) represents one of several variants of the stochastic approach, depending on the available mobility information. If a node knows nothing about the network's state, then all the node can do is to randomly forward packets to their neighbour(s).

If a node, however, has the ability to estimate the *forwarding probability* of some of its neighbours, then a better forwarding decision could be made. Protocols in this category are referred to as *History-based* or *Estimation-based*.

Another way to describe those schemes is to divide them into *source routing* and *per-hop routing* [22]. This classification is similar to the previous two protocols in many ways. In source routing, the complete path of a message is determined at the source node and encoded in the message. The route is therefore determined once and does not change as the message traverses the network. In contrast, in per-hop routing, the next-hop of a message is determined at each hop along its forwarding path. Per-hop routing allows a message to utilize local information about available contacts and queues at each hop, which is typically unavailable at the source. Source routing may appear both confusing and absolutely useless as it has been mentioned that source nodes in a DTN are clueless when it comes to finding end-to-end contemporaneous paths. The decisions made by source routing protocols, however, are based on some level of mobility knowledge and are bound to specific DTN applications, which will be described shortly.

Furthermore, if the mobility patterns can be used in the forwarding probability estimation, an even better decision may be made. Protocols in this category are referred to as *model-based* forwarding protocols [4].

In some cases, network efficiency can be achieved if the locations of certain nodes are controlled by moving them from one point to another. This approach leads to a sub-categorization of schemes into either *proactive* or *reactive* routing [22]. The former describes cases in which the movement of nodes is controlled. Routes are computed here automatically and independently of traffic arrivals, while the latter is for protocols in which movements of the nodes cannot be controlled, and when routes are discovered on-demand when traffic must be delivered to an unknown destination.

3.2 Mobility Knowledge

The knowledge of nodal inter-arrival times (i.e. when nodes enter each others' transmission premises) is essential for DTN routing. This is translated to the amount of link-state information available with respect to the mobility of the nodes in the DTN.

There could be either full knowledge, partial knowledge or zero knowledge of the networks topology; and this knowledge is absolutely dependent on the particular type of DTN application at hand. Thus, it is fair to say that we may map certain routing schemes to certain applications, or claim that a specific DTN routing scheme is *more suitable* for a certain DTN application type, based upon that application's available knowledge of nodes mobility, among other metrics (e.g. delivery and delay constraints). This mapping will be introduced towards the conclusion of this report.

In some DTN applications, the future behaviour of nodes may be almost fully known. This is especially true with bus routes and planetary trajectories, for instance, where the corresponding nodes in both cases (buses and satellites, respectively) move along strictly defined and scheduled paths. But this is not the case with many other DTN applications. One would assume that wild animals, military vehicles or personnel and common villagers tend to move in some random manner. This, however, is not exactly the case. In fact, it is widely believed that node mobility patterns (while random) are generally recurrent [20]. Some types of nodes tend to re-visit some previously visited locations over time. This will be further discussed in Section 5.

Recurrence is a common property of mobility models in DTNs. For example, humans tend to perform repetitive tasks (such as going to work, grocery shopping, entertainment, etc), workers often have repetitive responsibilities (running certain types of errands, meeting with specific clients, etc), and many mobile agents have a small set of frequently revisited destinations (cars revisit gas stations, birds return to their nests, animals frequent the same water sources, etc).

Recurrence is different from temporal locality in that a much weaker assumption is made in the former on *when* a particular event is repeated. Hence, cache replacement policies such as LRU will not perform well due to their direct emphasis on time of occurrence. Instead, replacement policies such as LFU should work better because they are a function of recurrence, not absolute time. Again, it is out of the scope of this document to discuss the details of queue management policies in DTNs.

In the next section, a detailed description of the main DTN routing schemes is presented. Each routing scheme will be presented according to some performance properties.

3.3 DTN Routing Schemes

3.3.1 Random and Epidemic Routing

Random routing (or *simple flooding*) is applied when zero-knowledge of the network's topology is available. If the node carrying the data bundle to be forwarded has no history data or knowledge of the mobility patterns of its neighbours or of the destination node, then the simplest decision is to send the message (or a copy of it) to each node it encounters within its transmission premises.

This, of course, will result in considerable redundancy in the network. The network's combined buffer space will drastically reduce in size since multiple copies of the same messages are being circulated over the network. In addition, link bandwidth is consumed by these re-transmissions.

In order to reduce this redundancy factor, a modified random approach, called *Epidemic Routing* (ER), checks before forwarding any message. If the neighbour node was not the destination node, and if it already has a copy of that same message, then the message is not forwarded. When two nodes are within communication range, they exchange pair-wise messages that the other node has not seen yet. Thus, as long as buffer space is available, and the node is not "immune" to the transmitted data (i.e. has not encountered its latest version yet) then messages will spread like an epidemic disease among nodes through a series of "infections" [1].

Certain studies [41][39] show that ER is capable of delivering almost all transmitted message. Such studies, however, assume unlimited, or sufficiently large, buffer sizes at each node, which is a non-realistic assumption. Random routing hence may result in halting the whole network if the amount of exchanged/redundant message over exceeded the actual buffer space limit. In order to contain this devastating by-product, a form of controlled simple flooding, known as *Spray and Wait* was introduced [39].

In Spray and Wait, a node performs a *controlled local broadcast* (a relay) only to its immediate neighbours. A source node initially spreads *L* number of copies over the network to *L* distinct relays (spray phase). If the destination is not found during this phase, each node that is carrying a copy of the message performs a direct transmission (i.e. it buffers the data and enters a wait phase until the ultimate destination is found) [1]. Note that such an approach requires a network with sufficient nodal mobility to succeed [48].

3.3.2 Direct Delivery

An alternative extreme routing approach is to let the source hold the message and deliver it to the destination only when they are within communication range [40]. This approach obviously has lower overhead, but the delay can be extremely long especially if the nodes have zero-knowledge of the network's topology. The worst case scenario may occur when the node carrying the data never gets to the range of the final destination and the message never gets transmitted.
This scheme, however, would serve the purpose perfectly in case of full knowledge of mobility patterns, i.e. if the path of the nodes is highly predictable or even predetermined. As an example: a train on the rail road is known to pass by a specific point in a specific time interval unless it is faced by an unexpected obstacle that would either further delay it, or unfortunately force it to change its rout.

3.3.3 Probabilistic/History-based Routing

As mentioned in Section 3.2, some types of mobile nodes are not likely to move around randomly, but rather move in a predictable fashion based on repeating behavioural patterns. If a node has visited a location several times before, then it is likely that it will visit that location again. This observation supports the need to find an alternative scheme to ER with lower demands on buffer space and bandwidth capacities, and with equal or better performance results.
Probabilistic (or history-based) routing is a deterministic approach that depends on a measure of delivery-likelihood per node. A well-known implementation of this scheme was introduced in [32] as PRoPHET: a Probabilistic Routing Protocol using History of Encounters and Transitivity.
PRoPHET and other history-based approaches generally utilize a probabilistic metric called *delivery predictability*, $P_{(a,\ b)}$ [0,1], that is established at every node *a* for each known destination *b*. It indicates how likely it is that this node will be able to deliver a message to that destination. When two nodes meet, they exchange summary vectors (as in ER) and also a delivery predictability vector containing the delivery predictability information for destinations known by the nodes. The information in the summary vector is used to decide which messages to request from the other node based on the forwarding strategy in use.
The calculation of the delivery predictabilities as introduced by [32] has three steps. First, the metric is updated whenever a node is encountered. This leads to higher delivery predictabilities for nodes that are more often encountered. This calculation is shown in Equation 6, where P_{init} (0, 1] is an initialization constant:

$$P_{(a,b)} = P_{(a,b)\ old} + (1 - P_{(a,b)old}) \times P_{init} \qquad (6)$$

It is only logical to assume that such metric would be affected by *age,* a queueing policy metric that was introduced in Section 2. A pair of nodes is less likely to be good forwarders of messages to each other if they don't encounter each other in a while, and their *P*-values must be reduced. The aging equation is shown in Equation 7, where γ *(0, 1)* is the *aging constant*, and *k* is the number of time units that have passed since the last time the metric was aged:

$$P(a, b) = P(a, b)\ old \times \gamma^{k} \qquad (7)$$

One important feature of the delivery predictability is its *transitive* property. If node *A* frequently encounters node *B*, and node *B* frequently encounters node *C*, then node *C* probably is a good node to forward messages destined for node *A*. Equation 8 shows how this transitivity affects the delivery predictability, where *β [0, 1]* is a scaling constant that decides how large impact the transitivity should have on the delivery predictability:

$$P(a, c) = P(a, c)\ old + (1 - P(a\ , c)\ old\,) \times P(a, b) \times P(a, c) \times \beta \qquad (8)$$

PRoPHET suggests a probabilistic routing environment in which the network resources are being preserved and managed in a manner that is more efficient than what is actually done under ER. The authors of [32] argue that it is more realistic in dealing with scarce buffer space.

Nevertheless, History-based Routing in general has its weak points. Particularly in DTNs, there is always a trade-off between message duplicity and latency. Although fewer copies of a certain message may result in a decrease of overhead, this, however, doesn't necessarily translate into higher delivery rates compared to Random Routing, for instance. If a message bundle is being held by only one mobile DTN node and this node happen to get destroyed due to any of the many extreme conditions in the DTN setting. Then, the bundle it carries is lost permanently.

In addition, the next hop choice in Probabilistic schemes –in general- is made only based on the probability that the chosen next hop encounters the bundle's ultimate destination and not on the possible encounters that it may have with other nodes on the way having higher delivery probabilities. Not to mention that such schemes make use of a learning process that takes a considerable amount of time in the presence of excessive delays. Last but not least, nodes are –still- not able to construct routing tables simply because what presents itself at a current instant as an opportunity might not be present again in the near future.

3.3.4 Model-based Routing

Model Based Routing (MBR) uses *world models* of the mobile nodes for a better selection of relaying nodes and the determination of a next hob's location without flooding the network [4] [19]. World models contain location information (e.g. road maps or building charts) and user profiles indicating the motion pattern of users.

In many real-world applications, considerable –almost full- knowledge of node trajectory is available. This fact may initially contradict with the view of a DTN established earlier. But such applications still suffer from the rest of DTN's strictly limiting characteristics. In these particular cases, mobile nodes move following strictly defined paths as in the case of walking along a street or driving on a highway [12]. Terrestrial objects bound to their orbits may be included under this category as well.

The key idea of this approach is to take into account that mobile devices typically do not follow the random walk motion pattern but are mostly carried by human beings. Once a description of moving pattern is provided by a *user*, then the intermediate nodes have a better judgement with much higher probability of when and where to forward the data packet towards the final destination.

Note that for the sake of performance evaluation, many mobility models have been developed. Those models are usually classified as *realistic* traces or *synthetic*. They are also described as being *entity* or *group* mobility models. The most popular according to some surveys [5] [7] [15] are *random walk* and *random waypoint*. Those particular models, however, don't serve the purpose of DTN applications since we are dealing with traces with much lower level of randomness. Mobility models, in fact, are also used to assist the evaluation of other routing approaches such as Probabilistic Routing [32].

3.3.5 Node Movement Control-Based Routing

Instead of leaving the mobile host in a state of passive waiting for a reconnection opportunity, it may be designed to actively modify its trajectory in a way that enhances the system's performance whenever possible (e.g. reduce the transmission delay of messages).
Li and Rus, among others, proposed in [30] a routing approach that involves the exploitation and deliberate controlling of node mobility. Hence, overall system's performance metrics, delay in particular, can be improved. Protocols in which the movements of nodes are controlled are also referred to in the literature as *proactive*, as oppose to *reactive* protocols in which the movement of the nodes is not controlled.
There are many variations of the proactive routing scheme, depending on the level of randomness introduced to its nodes mobility patterns. One class of movement-controlled routing is *Message Ferrying* [50] [51]. Message Ferries (MFs) are special mobile nodes with limited storage capacity that periodically transit from one location to another, carrying bundles between other disconnected stationary nodes in the network. In [49] twelve variants of the MF protocol are listed. Two of those are discussed here. These two schemes basically differ in whether ferries initiate non-random proactive movement. In the Node-Initiated MF (NIMF) scheme, ferries communicate with nodes they encounter while moving around the deployed area according to specific predefined routes. With knowledge of ferry routs, nodes periodically move close to a ferry and communicate with it. Here, since non-ferry nodes obtain knowledge of the ferries routes and proactively and periodically move to meet up with them. As the sending node approaches the ferry, it forwards its messages to the ferry that will be responsible for delivery. The other form of MF is referred to as Ferry-Initiated (FIMF). Here, ferries move proactively to meet nodes. When a node wants to send a data bundle to other nodes or receive packets, it generates a service request and transmits it to a chosen ferry using a long-range radio. Upon reception of a service request, the ferry will adjust its trajectory to meet up with the node and exchange packets using short-range radios.
Message ferrying, in general, is based upon utilizing a non-randomness factor to the mobility of the nodes in the network and exploiting this non-randomness to help deliver data. In contrast to this approach, *DataMULES* [37] is another proactive approach that adopts total randomness. DataMULES are mobile nodes with arbitrary mobility patterns and equipped with large storage capacities and renewable energy sources. The proposal consists of a three-tier architecture (sensor nodes, DataMULES and access points) and is supposed to connect spare sensors at the cost of high latency. In the DataMULES architecture, sensor nodes are expected to perform minimal work (i.e. mere sensing). Mobile DataMule nodes, however, are assumed to roam across the platform, read (or exchange) data held by sensor nodes via short-range wireless communication and to finally transmit the sensed/collected data to some access point in the premises. The main advantage of this approach is in major power savings for the sensor nodes in the network, which will consequently allow them to sustain longer life cycles. Communication in this setting takes place over a short ranges and is fully delegated to DataMULES that employ extra capabilities in terms of mobility, transmission, buffer-space and battery life.

3.3.6 Coding-based Routing

Erasure Coding [42] and *Network Coding* techniques [44] have been also proposed for DTNs. Erasure coding involves more processing and hence requires more power. However, it was shown to improve the worst-case delay in [42]. It is particularly useful when applications require bundles to be delivered within a specific time interval. The basic idea of erasure coding is to encode an original message into a large number of coding blocks. So if the original message contains k blocks, using erasure coding, the message is encoded into n ($n > k$) blocks such that if k or more of the n blocks are received, the original message can be successfully decoded. Here, $r = n/k$ is called the *replication factor* and determines the level of redundancy.

Network Coding comes from information theory and can be applied in routing to further improve system throughput. Instead of simply forwarding packets, intermediate nodes combine some of the packets received so far and send them out as a new single packet. For example, suppose that there are three nodes, A, B, and C. Nodes A and C want to exchange information through the middle node B. Node A first transmits packet x to node B, while node C transmits its packet y to node B. Node B then broadcasts a packet which the result of (x XOR y). Since node A has packet x, and node C has packet y, node A can decode y and node C can decode packet x.

Simulation results in [44] show that, for that particular given network setting, the packet deliver ratio using Network Coding is much higher than that under Probabilistic Forwarding, and most of the packets are delivered with a lower forwarding factor.

3.3.7 Vector-based Routing Schemes

By closely examining the literature, it can be fairly claimed that most of the significant DTN routing schemes that were proposed after 2007 are variants of the basic approaches mentioned above. Those variants attempted to overcome some of the limitations of their parent schemes in the hope to provide better routing in terms of delivery rate, congestion management, delay and other performance metrics.

Vector Based Routing (VBR) represents an example to support the argument above. It was actually proposed from two different perspectives as an augmentation to two main stochastic routing approaches: Flooding and History-based Routing.

In [25], the authors present a *Flooding-based Vector Routing* (FVR) protocol. As with all enhancement efforts to the flooding approach, FVR aims at reducing the number of message duplications in the network while achieving an acceptable performance in terms of delivery ratio and delays. Here, each node periodically computes a current vector V_{cur} by comparing its coordinates (x_t, y_t) and $(x_{\Delta t}, y_{\Delta t})$ at an instant $t - \Delta t$.

Nodes in FVR exchange as they meet vectors that include information about their direction and velocity. Each node makes a decision based on this information regarding the suitability of one of its currently available neighbours to act as a next-hop for a given message and also how many message copies should be replicated to each of those next-hops. The “vector factor” appears when FVR favours neighbours moving in a direction *orthogonal* to the movement of the deciding node. A larger number of message replicas will be forwarded to such nodes since they probably have better chances in meeting the destination. Nodes moving in the same direction and speed of a current node will, consequently, not obtain any message replica because they are probably never going to reach the destination before it.

FVR also relies on the velocities of the nodes in order to decide how many message copies to forward. Nodes moving in the same direction of the current deciding node may receive a certain number of message replicas only if those nodes have a higher velocity and are, as such, most likely going to reach the destination faster.

The same authors introduced the concept of vector routing again in [26] where they proposed a *History-based Vector Routing* (HVR) protocol. Here, each node in the network creates and manages its own location vector history while keeping a record of the vector information history of all its current neighbours as well. An exchange of vector information is then conducted between neighbouring nodes. As such, every node will own a database containing information about the location of all its neighbours. This database will be updated as new encounters occur. Each of these databases will therefore serve to perform more efficient forwarding strategies.

An obvious argument against VBR, however, is related to its dependence on rapid vector exchanges between adjacent nodes. This procedure conducted in both FVR and HVR may cause more harm to the network's performance since it will occupy more buffer space as the network becomes denser.

3.3.8 Other Routing Schemes

There are many routing approaches other than those mentioned above. It should be mentioned that all the schemes mentioned so far are unicast routing schemes. *Multicast Casting* [52] supports the distribution of data to a group of users. Many potential DTN applications operate in a group-based manner and require efficient support for group communication. One example in the DTN context is disaster recovery scenes, where it is essential to distribute critical information about casualties and possible hazardous events to rescue teams. However, since the network is often disconnected in a DTN layout, multicasting in DTNs is a considerably different and challenging problem. The multicast group membership management should be re-defined here. The design of multicast routing algorithms (when and where to forward) in DTNs is obviously more challenging and will not be covered extensively in this document.

In addition, there is *Inter-Region Routing* [49]. This approach may be best witnessed in scenarios such as disaster relief efforts, battlefields, and remote disconnected villages. Here, nodes form clusters (also called *regions*) such that a communication path exists between any two nodes within each cluster/region. Region boundaries are used as interconnection points between dissimilar network protocol and addressing families. The boundaries are defined by some metrics such as link delay or connectivity, error rates, addressing mechanisms and quality of service provisions. Major issues including naming, binding, route selection, protocol translation, and reliability control must be addressed in inter-region routing in DTNs.

Another scheme known as *Delegation Forwarding* (DF) was introduced in [11] to cope with ER's major limitation; the large number of bundle duplications needed to achieve reliable delivery with minimal latencies. DF is basically based on assigning *quality* and *level* values to every node in the network. The quality of a node can be quantized using a combination of various metrics (e.g. delivery ratio, delivery latency, buffer occupancy, power consumption, number of message replica, etc.). The level of a node is initially set to be equal to its quality. As two nodes meet, forwarding from one node to the other occurs only if the latter has a higher quality level than the forwarder's level. As the forwarding process successfully completes, the forwarder raises its level to the higher quality of the receiving node. Therefore, in contrast to typical flooding strategies, under DF the higher the node's level is increased the lower its likelihood to further forward bundles. This reduces the number of bundle replicas.

Before discussing the predominant DTN fields of application, we will return to the queueing policies and forwarding strategies discussed earlier in Section 2 in order to validate them against the routing schemes described above based on the following routing metrics: delay, delivery and overhead.

4. Queueing Policies and Routing Schemes

The queue management and forwarding policies discussed in Section 2 may appear as synonyms for the same process. But they are not. The simplest scenario that explains how they complement each other is when a node carrying a data bundle comes in contact range with another node that happens to be that bundle's final destination. Here, the bundle is forwarded regardless of the applied strategy and is also permanently deleted from queue of its carrier since there is no need for keeping a copy of that particular bundle. Otherwise, only a copy of that bundle maybe forwarded –depending on the strategy- and some decision is to be made regarding the remaining copy in the node's queue.

This decision depends upon the global goal of achieving a better routing performance for the network as a whole. Such a goal depends on the global knowledge available regarding the state of the network. This raises the exact same questions raised in relation to routing schemes above. And it is only natural to conclude again that such "global knowledge" is highly unlikely, if not impossible, to attain in most DTN instances.

It is difficult then to compare routing schemes and queueing policies. This claim is supported by the fact that the studies proposing buffer management schemes often do so with respect to a specific routing approach or application field. The following subsections survey these management policies according to their corresponding routing schemes.

4.1 History-Based Routing

The queueing policies mentioned in Section 2.1: *FIFO*, *MOFO*, *MORP*, *SHLI* and *LERP* where proposed by [31] within a simulation study that involves Probabilistic Routing. The authors here conclude that probabilistic routing coupled with a wise choice of buffer management policies and forwarding strategies significantly improve the performance of DTNs in terms of delivery rate, overhead and delay. This is a perfectly natural assumption since probabilistic routing incorporates the learning process necessary to develop the knowledge about the forwarding frequency. The whole notion *of delivery predictability* is originally and exclusively related to probabilistic routing as opposed to many of the other routing schemes, since the evaluation of such a parameter depends on some past history of the encountered node. Nevertheless, probabilistic routing is not always the most suitable schemes and it does have its weaknesses as will be discussed later.

4.2 Epidemic Routing

The authors in [53], on the other hand, concentrate on the Epidemic scheme (ER) and argue that to propose buffer management policies such as Drop Tail (DT), Drop Front (DF), Drop Oldest (DO) and Drop Youngest (DY) employed by ER are not suitable for DTN overhead constraints. They instead propose a joint DTN scheduling and drop policy: Global knowledge Based

Scheduling and Drop (GBSD) that aims at optimizing two performance metrics namely: the average delivery rate and delay. GBSD primarily relies on global network information. The authors realize that such information is typically unavailable in DTNs. GBSD is therefore practically non-implementable and its value remains as just pure theoretical reference!

The work in [56] proposes a congestion control strategy called N-Drop (ND) to be used with the widely used ER. The authors consider a network of *M* nodes each of which has a FIFO buffer of size *L*. The queued non-fragmented bundles are transmitted one by one upon the successful pairing with another encountered node.

Upon the arrival of a bundle to a congested node, ND scans the buffer of that node in search for all the bundles that have been forwarded a number of times greater than or equal to a threshold *N*. All such bundles are then discarded and the new arriving bundle is inserted into the buffer. If on the other hand no such bundles were found, the bundle at the rear end of the queue is dropped and replaced with the newly arriving one.

Moving bundles back and forth from a node's buffer to its persistent storage and vice versa adds considerable processing overheads. This is especially true in challenged environments where bundles may be repeatedly forwarded until they are ultimately delivered. Therefore this solution to overcome disruptions and delays as proposed by the Bundle Protocol turns out to be inefficient in relatively extreme scenarios.

4.3 Node Movement Control-Based Routing

The authors in [55] investigate buffer allocation fairness in sparse Message Ferrying DTN (MF-DTN) where communicating nodes do not contend for wireless channel access. Instead, it is the traffic flows that contend for Message Ferries' buffer space. Each MF maintains information such as: visiting node sequence, session (i.e. flow) source/destination, per session and total buffer space allocation, session lifetime, and expected contact time with each visited node.

MFs use such information to construct forwarding tables, make forwarding decisions, and perform buffer allocation for admitted sessions. Stationary nodes maintain local session requests lists and their durations. Upon contact establishment, an MF retrieves such information from static nodes and uses it to update its global request list, determine all forwarding possibilities for each active session and select the best one according to a given metric. The authors define two performance metrics: Path Metric (PM) and Ferry Transportation Cost (FTC) based on which they tailor a buffer-based max-min fairness model. Using this model, they design a Fair Buffer Allocation Scheme (FBAS) and integrate it into a Buffer Efficient Routing Strategy (BERS) that decides on the suitable type of forwarding to take place based on PM.

4.4 Other Management Policies

Some buffer management policies are application specific. The work in [54] proposes a Buffer Management Policy for Mars *Intelligent Proximity Networks* (BMP-MIPN) used for the transfer of in-site sensing data. The authors here developed a buffer control strategy where image data streams are prioritized based on their significance in the taken image. More significant data segments are given higher priorities than those with lower significance. Moreover, bundles are ordered in descending order of priority with the most significant bundle being on top of the buffer. As forwarding opportunities become available, only the bundle at the front of the queue is forwarded. Any newly arriving message to a non-exhausted buffer-node will be inserted into the buffer in a way that is consistent with its priority level. If the buffer is full, then the arriving

bundle will be discarded if it has the lowest priority or will replace one of the buffered lower priority bundles.
This policy ensures that: a) high priority bundles are forwarded quickly, b) low priority ones are dropped as soon as there is a need and c) moderate priority bundles spend a considerable amount of time in the buffer and their status depends on priority characteristics of subsequent messages.
Since this particular policy is proposed to IPNs, it may be also mapped to direct delivery or Model-Based routing schemes. This hover is further elaborated upon in Section 6.

Table 1 below, summarizes the relations between the buffer management policies and routing schemes discussed above and lists the performance metric(s) emphasized upon by each buffer management approach:

Buffer Management Policy	Routing Scheme	Metric(s)
FIFO, MOFO... etc. [31]	History-Based	Delivery rate
DT, DF... etc.	Epidemic	None suitable for DTNs
GBSD [53]	Epidemic	Delivery rate and delay
ND [56]	Epidemic	Disruption and delay
MF-DTN [55]	MF	Path Metric and Ferry Transportation Cost
MIPN [54]	Direct Delivery, Model Based	Delivery of higher priority data

Table 1: Relations between the buffer management and routing schemes

5. Predominant DTN Applications

There are many real-life applications where wireless nodes, –mobile or stationary-, are forced to undergo extreme operational conditions and/or wait for extended intervals of time that exceed traditional IP forwarding times (that are usually measured in milliseconds) before being able to forward their data to next hops. In such applications, there is an essential need to provide an integral framework to connect nodes operating different standards. Nodes would also be required to adopt the SCF method. Efficient storage management techniques and specific routing/forwarding policies need also to be incorporated into such applications.
In the reminder of this section, some of the predominant DTN applications, according to the relevant literature, are listed and briefly discussed in terms of node mobility, delivery rate, levels of harshness and other metrics.

5.1 Wildlife Monitoring

As its name implies, this application is concerned with gathering data about wild faunae species and their habitats. Monitoring is conducted by attaching a sensing device to each animal (i.e. mobile node). These devices may contain microcontrollers, global positioning systems (GPS), orientation and temperature sensors, off-chip flash memories, RF modules, processing units and batteries (with solar modules to recharge) to provide power [34]. Such a device will turn the animal carrying it into a mobile node that gathers data about the rest of the herd's behaviour and track its geographical movement. Usually, the nodes will exchange data among themselves until they come in the vicinity of an *Info-station*, which is a base-station that collects data from these nodes. A mobile agent may also act as an info-station by driving through the animals groups and entering the ranges of the device attached to it. The info-station's mobility is suggested to be manually controlled.

One of the most famous examples of wildlife monitoring is the *ZebraNET* project [16] [47] conducted in Sweetwaters Reserve, Kenya. Here, zebras are equipped with custom tracking collars (nodes) and carried across a large wild area. The collars include GPSs, flash memory, wireless transceivers, and a small CPU. They operate as a peer-to-peer network to deliver logged data back to researchers. Since no base-stations are available in the vicinity of the experience, wildlife researchers drive through the herd's area collecting information about the dispersed zebra population.

Figure 4 - A zebra equipped with a ZebraNET tracking collar. Source: http://www.peizhang.com

The observation of the animals' behaviour in this particular example reveals a low level of randomness in their mobility. In turns out that zebras do follow some predicted mobility model. For instance, the whole herd tend to gather at the water source during some time of the day. Such observations help on deciding an appropriate routing approach among the ones mentioned in the Section 3.

In the same context, *Shared Wireless Info-station Model (SWIM)* is a project that aims toward gathering information about populations of whales and other marine mammals [38]. Here also, radio frequency devices are implanted or attached to the animals. Different types of whales have known typical dive times; times during which they remain underwater without surfacing. After several dives, the whales socialize and feed near the surface of the water for minutes or hours. Some whales are known to return to the same feeding grounds at regular intervals. These grounds offer proper locations for the placement of offloading radio stations. Underwater scenarios, however, are usually more complicated and are further discussed next.

5.2 Forestry and Underwater Sensors

In many areas, environmental monitoring is required for many reasons. Measurements regarding temperature, air pressure, intensity of natural lighting, chemical contamination in the soil or the water, fire hazards, radiation levels and other measurements can be gathered via special wireless sensors in both forestry and underwater applications. In such cases and due to the nature of the location, nodes (mobile or stationary) may experience long partitioning periods before contacting each other or some info-stations.

In forestry applications, sensor nodes may be lost (and hence the data they attain) due to many natural harshness factors including heavy rain, hail or snowfall and extreme temperature variations, among others. Nodes may also be physically destroyed by wildlife, or may fail due to hardware/software malfunctioning. Even the connectivity links between unharmed nodes are attenuated or lost due to natural causes such as dense trees. [2]

What is special about underwater sensing in particular is the medium of transmission (i.e. water). RF communications generally do not work in water. A wireless device using the 802.11b/g standard, for example, will have a radio range measured in centimetres. The best technology alternative for wireless underwater networking is *acoustic* networking, where sonar is basically used to carry data signals. However, this sonar–based physical layer has similar characteristics as the radio frequency transmission, with fading, multipath reception, reflections (off the sea surface and sea bottom) and so forth. [16]

In forestry applications, long-range radio transmission is usually deployed to forward data to the mobile base-station(s). Long-range radio overcomes the portioning limitations despite its negative effect on the nodes' battery-life. This approach, however, is not applicable in underwater sensor networks because the radio signal will be deviated or weakened in the water and will probably be lost before reaching the next hop.

In such cases, two options are available, either the underwater nodes are set to be mobile and are triggered to float to the surface and transmit their data contents to a nearby info-station Another option is to relay on a *DataSeal* (in analogy to the DataMule) which is built capable of submerging and vesting each underwater node by getting close enough to ensure a successful wireless transmission of data. The DataMule approach, naturally, is also favourable in forestry applications.

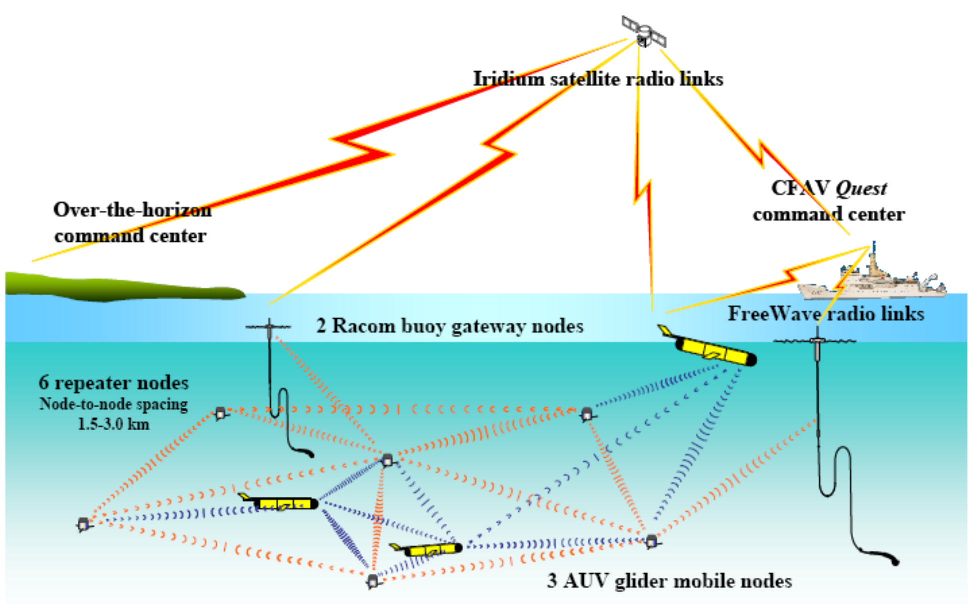

Figure 5 - Node Diagram of a SeaWeb network (from [35]).

Within the same context of underwater sensing, there is the example of *SeaWeb* [35], a project conducted by the U.S. Navy. SeaWeb has developed its own MAC layer protocol, especially optimized for the acoustic networking case. Figure 6 is a diagram of such a network showing how SeaWeb concept encompasses links to more standard naval communications via buoys can provide network connectivity for autonomous underwater vehicles (AUVs) and uses repeater nodes. The nodes in the SeaWeb have been mainly disposable in that they run on batteries with no possibility for recharging [16]. The diagram further implies the use of satellite signals to enhance broadcasting options, which is always a trend in DTN architectures that are meant to support networks' heterogeneity.

5.3 Village Networks

Village networks represent a very promising public application for DTNs, especially in secluded areas lacking communication infrastructure [17] [22]. Rural buses may be also used to provide Internet connectivity to isolated and remote villages [3] [36]. Busses act here as relays or couriers, transmitting and exchanging data via simple wireless transmission across the city's bus network. The transmission window here is narrow and limited by the time busses meet while in-route or drive across some info-station. The advantage of this approach, though, is in the fixed routes busses take in their routes across their network, which almost eliminates any randomness and guarantees a successful forwarding rate (i.e., data bundle is guaranteed to reach destination, given a much longer travelling time).

Another relevant example is the *Wizzy Digital Courier* service [45] that provides disconnected Internet access to students among other users in remote villages of South Africa. Here, a courier on a motorbike, equipped with a USB storage device, travels from a village to a large city that has high-speed Internet connectivity.

Typically, it takes a few hours for the courier to travel from the village to the city. But this approach has an obvious advantage over even the conventional Internet: the bandwidth is considerably larger than provided by the village's network. Here the amount of data that may be transmitted is only limited by the amount of hard-drives the courier can carry on his motorbike. This, according to today's technologies may reach several Terra bytes, an amount that cannot be matched by the most advanced wired networks nowadays. Like in the SeaWeb example, several other connectivity options may be integrated here (e.g. satellites, LEO, GEO, telephones) to aid the delivery process.

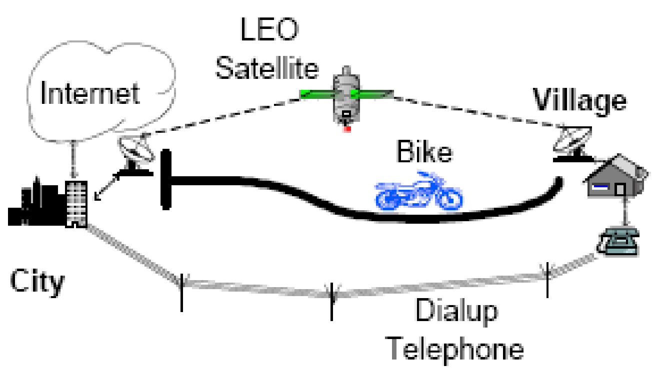

Figure 6 - A variety of connectivity options between a remote village and a city (from [22]).

5.4 Inter-planetary Networks

The massive distances separating terrestrial artificial objects and the need for these objects to exchange data among each other or with base-stations on earth –or perhaps other planets- represent an extreme case of DTN communication. This is physically known as: Light-trip times.
Ironically, it was NASA's vision of an Inter-Planetary Network (IPN) [9] that initiated the search for a heterogeneous architecture that overcomes the traditional limitations of TCP, which eventually evolved into the DTN field of research. Figure 8 illustrates how the first IPN was pictured at the time.
We may imagine a situation where a scientist on Earth is responsible for the operation of a robotic meteorological station located on Mars. If the scientist wants to upgrade the software in the weather station's data management computer by installing and dynamically loading a new module, then the module must be transmitted first from the scientist's workstation to a deep space antenna complex, then from the antenna complex to a constellation of relay satellites in low Mars orbit and finally from the relay satellites to a station on Mars [6]. The combination of long signal propagation times and intermittent connectivity caused by the interposition of a planetary body between the sender and the receiver can result in round-trip communication delays measured not in milliseconds or even minutes but in hours or days.
Despite those limitations, the extra-terrestrial networks may be compared to bus networks in terms of having couriers (satellites and busses, respectively) that follow fixed trajectories and pass by certain points within fixed time intervals. This would limit the routing problem to its minimum since each packet/bundle can predefine its path to the destination and the nodes would conduct forwarding decisions based on predefined routing maps.

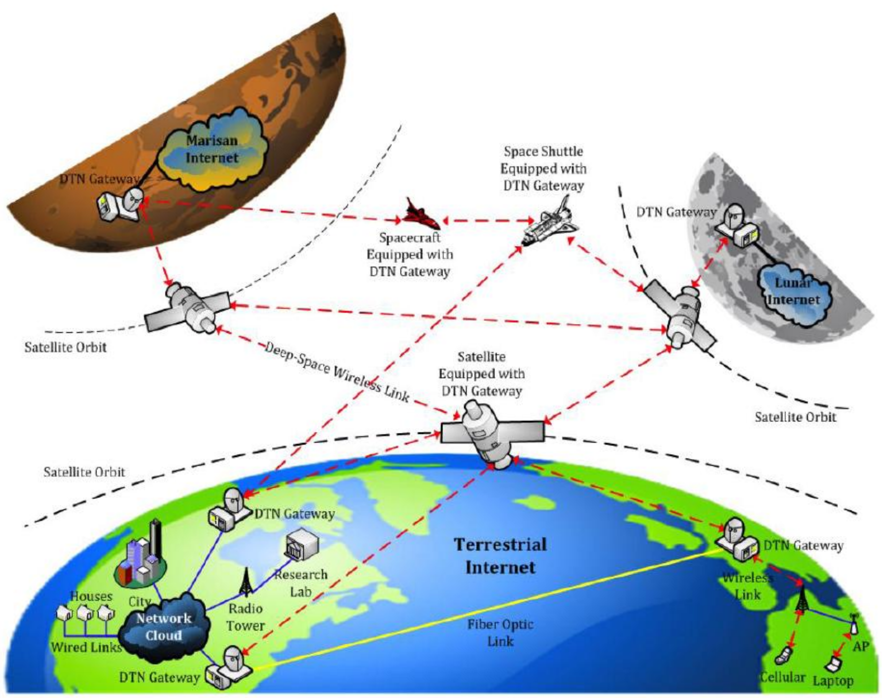

Figure 7 - Illustration of first IPN architectural definition as devised by [9].

5.5 Military Applications

Interestingly, all the potential DTN applications mentioned above may be also included, in a way or another, under the umbrella of military applications.

In military, as in the case of wildlife monitoring, there is a need to monitor extended geographical planes their objects and inhabitants –i.e. soldiers- who would be equipped with wireless sensors in order to indicate their locations. Also, the military would rely on a vast network of Ad Hoc wireless sensors, mobile or stationary, that are deployed over the battlefield's ground and water surfaces. Measurements similar to those conducted in forestry applications, related to detecting motion, change of atmospheric readings or chemical contamination are necessary. The same can be said about underwater sensors as seen in the SeaWeb project. As for village networks, we may assume that military vehicles will conduct the same role of mobile couriers in modern isolated battle locations, not to mention the dependency on satellite communications in modern warfare.

There are, though, major differences between the military application of DTNs and the previous ones: the fatality factor. Battlefields are dangerous. While it would be acceptable to assume human interference in collecting data from the nodes (e.g. an info-station is driven close enough to zebra herds to allow for wireless interaction), it is expected that a higher level of automation is presented in military application. Another important factor to consider is the node-loss ratio. Many nodes are to be destroyed in action. Redundancy in node deployment is, hence, required and the routing scheme would be much more complicated since it is expected to adopt several approaches to overcome the connectivity limitations. [16]

Tactical military networks are established in a very Ad Hoc manner. The nodes are in continuous and rapid motion. And there is most likely no stable infrastructure due since such infrastructure would just provide a target for the enemy. Network partitions, thus, are more likely to occur compared to other DTN applications.

There are other DTN application disciplines including industrial monitoring and disaster recovery (either being natural such as volcanic eruptions, hurricanes or earthquakes or man-made disasters such as terrorist attacks and car accidents). In all of these examples, wireless sensor nodes are simply required to periodically gather data related to the environment (e.g. heat, wind velocity, humidity, surface vibration, light intensity, noise, etc.) and report any extraordinary changes in their readings. These applications, however, do share many attributes and characteristics with the ones mentioned above and will not, hence, be discussed in further details here.

6. Summary and Discussion

This document aims towards creating a mapping between DTN routing schemes and DTN applications. As discussed in the last section, there is a variety of applications that share several attributes including extended transmission delays and intermittent loss of connectivity due to challenging topological circumstances.

Table 2 summarizes the properties of the DTN applications mentioned in Section 3 in terms of the nature of the mobility of the nodes and delay:

Application	Description	Properties
Wildlife Monitoring	Wireless sensor nodes attached to animals within a herd/flock. Wild-life researchers drive through a forest collecting information on fauna population	Animals' mobility is generally deterministic. Mobility plain is contained. Node position may be predicted based on some *recurrence*
Inter-Planetary NWs	A set of satellites or terrestrial objects in orbit exchanging data with each other or earth station	IPN consists of both terrestrial and interplanetary links, which suffers from long delays and episodic connectivity. Very long delays due to extreme distributed topology. Path is fixed though and will follow a known non-random mobility model
Village NWs	Buses or commuters act as mobile couriers serving stationary/mobile nodes along their route and relaying data between isolated locations	In case of bus NWs, routes are fixed and a mobility model can be set. In case of bikers, routes may be altered to serve further more locations. Advantageous on the side of bandwidth capacity
Underwater Sensors	Sensor nodes are being deployed in underwater environments	Long delay depending on visiting of Data Mule/Seal. Energy concerns due to need for nodes to surface. Water medium poses serious wireless transmission challenges
Battle-Field NWs	Wireless nodes are distributed in Ad Hoc topology in battlefields	Varying delay depending on collecting methods. Nodes may be mobile or stationary. High nodal loss ratio

Table 2: Properties of DTN applications.

There are, however, some particular characteristics that may allow us to create some clear-cut classification criteria among these DTN applications and to group each of them under some main predominant category. The creation of such classification method represents the first step towards this report's goal.

One way to approach this is to realize that DTN applications may differ according to their nodes capabilities in mobility, buffering and transmission. There is also the level of randomness in motion patterns and the frequency over which network partitioning occurs, not to mention the nodes' proposed duty cycles, the type of communication medium and the amount of data to be delivered and urgency in which this has to be performed. These are all DTN characteristics to be considered when attempting to categorize DTN applications.

In order to suggest some classification division for predominant DTN applications, we conduct the following mapping in Table 3 between a number of common applications mentioned throughout this report and some critical DTN characteristics and attributes:

	Stationary Sensing Nodes	Random Mobility Sensors	Deterministic Mobility Patterns	Higher Node-Loss Ratio	Physically Challenging Transmission Medium	Higher Level of Partitioning	Need for MFs/ Curriers	Delay is Significant	Buffer Constraints	Energy Constraints
Forestry	✓			✓		✓	✓		✓	✓
Wildlife Monitoring		✓	✓	✓		✓	✓			✓
Natural Disasters Sensing	✓	✓		✓	✓	✓		✓		✓
Underwater Sensing	✓	✓			✓	✓	✓		✓	✓
Inter-Planetary Networks			✓		✓	✓		✓	✓	
Village NWs	✓		✓			✓	✓			
Bus NWs	✓		✓				✓			
Personnel Monitoring			✓							
Accident/ Disaster Recovery	✓	✓		✓		✓		✓	✓	✓
Industrial Sensing	✓	✓		✓	✓			✓		

Table 3: Mapping DTN fields of application to network characteristics.

The networking attributes stressed upon in Table 3 are mainly related to nodal mobility patterns, latency/delay constraints, the possible aid of courier agents (e.g. Message Ferries), in addition to the level of harshness in the sensed environment. These attributes were selected based on the accumulative discussion over the previous sections.

For the sake of clarity, some points presented in Table 3 deserve to be commented upon:

i. We see that some applications (e.g., natural disaster sensing) were described as having both stationary and mobile sensing nodes. This is a normal situation in a DTN setting that would integrate many sensing platforms under one networking solution/application.

ii. Similarly, some applications (e.g., wildlife monitoring) may have nodes that experience both random and deterministic mobility. This is, again, expected in case of wild animals that may never stick to the concept of recurrence for many reasons.

iii. In the table above, the attribute: *Physically Challenging Transmission Medium* refers to circumstances where the wireless signal emitted by the sensor node may be deviated or weakened due to the surrounding medium. This includes water, vapor, volcanic ash, industrial fluids and light-trip times (in terrestrial applications).

iv. It is obvious that almost all the applications listed in Table 3 share the *High Level of Partitioning* attribute. This represents no surprise and only restates a definite feature of DTN applications in general.

v. As for the *Significant Latency* attribute, it is interesting to observe that only few applications were assigned to it. "Significance" is defined here according to the necessity (in terms of time-intervals) the data bundle has to be forwarded or delivered. This is the case in IPNs, for instance, since failure in seizing the opportunistic forwarding window would cause an additional indefinite delay –or possibly the permanent loss- of the bundle. Significance is also crucial when the data concerns life/property threatening hazards. Such as the case with disaster recovery applications.

As for the routing schemes, Table 4 lists the main properties along with major weaknesses of each of the schemes mentioned in Section 3. The emphasis here is on the amount of available knowledge regarding the mobility patterns of the nodes, redundancy buffer and delivery constraints.

Routing Scheme	Properties	Downsides
Direct Delivery	Guaranteed delivery if the destination node is encountered. Maximum delay expected	Carrier node may never meet ultimate destination and data bundle will be delayed indefinitely
Deterministic	Only used when future topology is fully/partially known. Otherwise: SCF	Mostly not the case with the predominant DTN applications
Random/Epidemic	Almost Guaranteed to deliver data bundle to destination. Only option if no mobility knowledge is available	Generates considerable overhead and requires sufficiently larger buffer size per node. Broadcast storms cause congestion
Probabilistic/History-based	Overhead (data redundancy) is considerably less than random routing. Quite popular in relatively smaller networks	Learning process takes long time in bigger networks resulting in excessively high delays. Lesser copies of a given message are available at a given time which raises concerns on permanent data loss
Model-based	Applicable if there is a life-model about the topology	Some knowledge of motion patterns have to be available. Otherwise of no use
Movement Control-based/DataMULES	If the movement of the nodes can be controlled (proactive protocols). Otherwise, an info-station may move randomly to collect data from the nodes. Limited Delays	Special mobile nodes are required with special storage. Access point architecture required
Coding-based	In particular settings, provides higher delivery rates than Probabilistic schemes.	Nodes are required to perform coding operations resulting in additional energy consumption. Transmission of forwarding factors further consumes BW and buffer space

Table 4: Summary of main DTN routing schemes.

Again, several items in Table 4 require additional commenting and observations gathered from the literature and some conducted simulations:

i. Note that both DataMULES and Message Ferries are considered in table 4 as routing schemes and not as tools to facilitate data forwarding between nodes in a DTN network. This decision is actually supported by some studies [48] although MFs were attractive factors in many DTN scenarios mentioned in Section 4.

ii. Certain simulation results [41] show that ER is capable of delivering almost all transmitted messages (89.9%). Those simulations assume, however, unlimited buffer sizes at each node, which is a non-realistic assumption. This is a non-realistic assumption, of course. Neither bandwidth nor buffer space is infinite, but instead they are rather scarce resources, especially in the case of sensor networks.

iii. There is always a trade-off between throughput and energy consumption.

iv. The energy consumed by the flood-based protocol can be eight times that of the history-based protocol.

v. Flooding makes sense at low-radio-range and low-connectivity points in the design space.

vi. Simulation notes show that for the network considered, the improvement of packet delivery ratio under PRoPHET over the ER can be up to 40 percent [32].

vii. Regarding reactive routing protocols, it is obvious that instead of letting the mobile host wait passively for reconnection, the mobile hosts may actively modify their trajectories to minimize transmission delay of messages.

When comparing DTN routing schemes, it has been the trend in the literature to base these comparisons on several performance metrics including: delivery ratio, delivery latency, buffer occupancy, power consumption, number of message replicas and others [32, 41, 48, 49]. The argument of this report, however, lays in the claim that *it is not enough to include such performance metrics alone. DTN routing after all, as shown throughout the discussion above, is very much an application-dependent order*. It is the specific application along with its precise circumstances in terms of nodal mobility, harshness factors, topological topography, level of partitioning, urgency of transmitted data and others. Application-specific-attributes such as these and the ones listed previously in Table 3 should be responsible for finalizing the decision on choosing a proper DTN routing scheme to be adopted for a specific DTN application.
Accordingly, in this document, potential DTN applications are broadly classified into the following types:

1- Sensing Applications in Harsh Environments: Including industrial locations, volcanoes, forests… etc.
2- Life-Monitoring Applications: This includes nodes attached personnel in workplaces as well as to wild animals.
3- Vehicular Mobile Networking Applications: Includes Village Networks, Buss Networks and networks populated by vehicles in metropolitan topologies
4- Military Applications.
5- Underwater Sensing Applications.
6- Inter-Planetary Networks.

To sum up the mapping effort, Table 5 indicates the suggested relation between the six classes DTN applications mention above and the DTN routing schemes based on the metrics listed above. It is worth observing that in almost all the applications, random flooding can be avoided and is probably replaced by another routing scheme. It is only in the military applications that flooding is suggested but only as a last resource given the fatality factor mentioned earlier. It is surprising that Random/Epidemic flooding has been always suggested as a preferred routing method despite the fact that it consumes the power/processing resources dramatically. Other less-stochastic approaches may be as sufficient as Epidemic flooding given certain delivery constraints.

	Direct Delivery	Random Flooding	History Based	Model Based	Controlled/ DataMULES
Applications in Harsh Environments	Poor	Fair	Poor	Good	Fair
Inter-Planetary NWs	Good	Poor	Good	Good	Fair
Life-Monitoring Applications	Fair	Poor	Good	Good	Fair
Vehicular Applications	Good	Poor	Fair	Good	Fair
Underwater Sensors	Fair	Poor	Poor	Fair	Good
Military Applications	Good	Good	Good	Good	Good

Table 5: DTN application classes mapped to routing schemes.

The three classification levels in Table 5 are based upon the following evaluations:

- Good: Routing scheme is suitable for delivery constraints, level of nodal mobility and delay constraints.
- Fair: Routing Scheme is not the best option. But may provide satisfactory delivery results in ideal scenarios.
- Poor: Routing scheme is not suitable for delivery constraints, level of nodal mobility or delay constraints.

We may notice for instance that all routing schemes are recommended to be used with military applications. This is because all schemes are usually implemented in battlefields and all mobility patterns are as well. The vital nature of the data to be delivered imposes this openness in implementation.

Direct delivery is marked Poor for harsh environment applications since it would not be applicable to wait until the destination node is found in order to report the incident of a forest fire.

History-Based routing is marked “Fair” for Vehicular application because it does not provide the flexibility provided by model-based. There may never be a way to predict which route a driver may take in given new topology (city), for instance.

7. Conclusion

Delay Tolerant Networks provide an integral architecture allowing partitioned heterogeneous networking instances to communicate in an otherwise disconnected setting. In many situations, wireless networking is challenged by intermittent connectivity and long lasting partitioning that yield indefinite transmission delays further exceeding threshold limits defined by TCP/IP standards.

Traditional Internet solutions are, hence, incapable of guaranteeing connectivity to networks of wireless sensors laid out in harsh areas such as forests, deserts, industrial plants, underwater habitats, natural disasters and accident scenes, battlefields and IPNs where sensor nodes are to experience, in addition to the challenges mentioned above, hazardous factors that may result in their own demolition and, consequently, the data they carry.

Through a Store-Carry-Forward approach, DTN wireless nodes are capable of buffering replicas of the messages circulated around the network until a suitable forwarding opportunity appears. Forwarding, in the DTN context, is interchangeably used with routing. Routing in the usual end-to-end sense is not quite applicable under DTNs since there is no way for a node to guarantee –or know about- a full contemporaneous path to the intended destination due to the intermittent connectivity nature of the situation.

This exceptional nature caused an outpouring of research attempts to design DTN-specific routing schemes. These schemes are mainly classified as being either deterministic or stochastic, based on the available knowledge of the mobility patterns of the nodes. Mainly and because very little of such knowledge is usually available, routing/forwarding in DTNs tend to be mostly conducted either randomly, or according to some probabilistic function that is generated over a long history-based learning process. Each of the two approaches has its pros and cons. It is sufficient to mention here that there always exists a trade-off between bandwidth/buffer space and latency when it comes to those two schemes in particular, and that rival routing schemes seek to walk the line between those two performance metrics, among others. Alternative DTN routing schemes that were also discussed in this report include model-based, movement-control-based, coding-based or vector based.

When compared to each other, routing schemes are judged according to specific performance metrics including: delivery ratio, delivery latency, buffer occupancy and power consumption. Published simulation results tend to favour some schemes over others. But those results are mainly biased by the assumed network settings in their corresponding simulations, or by some non-realistic assumption such as the availability of infinite buffer spaces. Surveying real-life DTN application examples reveal, however, that routing schemes tend to be application-specific rather than dependant on some delivery metric. That is to say, each and every application scenario has its own network characteristic requirements that will finally decide on the routing scheme to be chosen.

Since buffer management is an essential component of any DTN system, we tried to link some proposed queueing policies with common routing schemes. Our comparison proposes that indeed, some buffer management policies are more suitable for specific routing schemes in term of metrics such as delivery and delay.

The argument that was discussed over the length of this report lays in the claim that it is not enough to include delivery performance metrics alone. DTN routing, after all, is very much an application-dependent order. It is the specific application along with its precise circumstances in terms of nodal mobility, harshness factors, topological topography, level of partitioning, urgency of transmitted data and others.

In the last section of this report, an attempt was conducted to come up with broad classification bases for a predominant DTN application, and to map these to the major DTN routing schemes. This attempt was based on the application-related networking attributes mentioned earlier.

This effort, of course, is incomplete. It lacks the numerical analysis that supports its claims. In addition, there are many other DTN routing schemes that were not discussed in this report. This is mainly because of two reasons: First, it is extremely difficult to completely cover such a vast area of research in a report of this magnitude and addressed to its particular academic purpose. Second, it is our belief that the schemes covered above are the most essential in the literature, in the sense that most of the later schemes are basically variations of theses mentioned in Section 3 of this report. Thus, the proposed classification and mapping approaches may be enhanced as additional analytical results are involved and a further in-depth review of emerging DTN routing schemes is conducted. We aim toward expanding in this direction in the followings stage of this work.

References

[1] M. Abdulla and R. Simon, “The impact of the mobility model on delay tolerant networking performance analysis,” in *40th Annual Symposium in Simulation (ANSS'07), 2007*, pp. 177-184.

[2] F. Al-Turjman, A. Alfagih, H. Hassanein and M. Ibnkahla, “Deploying Fault-Tolerant Grid-Based Wireless Sensor Networks for Environmental Applications,” in *Proc. of the IEEE International Workshop on Wireless Local Networks (WLN)*, 2010. (Accepted).

[3] A. Balasubramanian, Y. Zhou, W. B. Croft, B. N. Levine, and A. Venkataramani, “Web Search from a Bus,” in *Proc. of the second workshop on Challenged networks, CHANTS*. ACM, 2007.

[4] C. Becker and G. Schiele, “New Mechanisms for Routing in Ad Hoc Networks,” *4th Plenary Cabernet Workshop*, Pisa, Italy, Oct. 2001.

[5] C. Bettstetter, "Mobility modeling in wireless networks: categorization, smooth movement, and border effects," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, 2001, pp. 55-66.

[6] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott and H. Weiss, "Delay-tolerant networking: an approach to interplanetary internet," *IEEE Communications Magazine*, vol. 41, 2003, pp. 128-136.

[7] T. Camp, J. Boleng and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Communications and Mobile Computing*, vol. 2, 2002, pp. 483-502.

[8] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall and H. Weiss. “Delay-Tolerant Network Architecture,” RFC 4838, 2007.

[9] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, E. Travis and H. Weiss, “Interplanetary Internet (IPN): Architectural Definition,” http://www.ipnsig.org/reports/memo-ipnrg-arch-00.pdf.

[10] T. Clausen and P. Jacquet, “Optimized Link State Routing Protocol (OLSR),” RFC 3626, IETF Network Working Group, October 2003.

[11] X. Chen, J. Shen, T. Groves and J. Wu, “Probability Delegation Forwarding in Delay Tolerant Networks,” in Proc. of the *18th IEEE ICCCN ‘09*, San Francisco, USA, August 2009.

[12] Z. Chen, H. Kung, and D. Vlah. “Ad hoc relay wireless networks over moving vehicles on highways,” In *MobiHoc 2001*, October 2001.

[13] Dang, H. and Wu, H. “Mobility Models for Delay-Tolerant Mobile Networks,” *The Third International Conference on Sensor Technologies and Applications*. *SENSORCOMM 2009*. June 18-23, 2009 - Athens, Greece

[14] A. Di Nicol`o and P. Giaccone, “Performance limits of real delay tolerant networks,” in *IEEE WONS*, 2008, pp. 149–155..

[15] E. S. Elmallah, H. S. Hassanein and H. M. AboElFotoh, "On the use of a simple mobility model in ad hoc routing," in *Proc. of International Conference on Parallel Processing Workshops,* 2001, pp. 479–484.

[16] S. Farrell and V. Cahill, "Delay- and Disruption-Tolerant Networking," Artech House INC., 685 Canton Street, Norwood, MA 02062, Forewordp. ix, 2006.

[17] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proc. of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications,* 2003, pp. 27-34.

[18] J.J. Garcia-Luna-Aceves, C. Marcelo Spohn, "Source-tree routing in wireless networks," in Proc. *of the Seventh Annual International Conference on Network Protocols*, Toronto, Canada, October 1999, p. 273

[19] J. Ghosh, H. Q. Ngo and C. Qiao, "Mobility profile based routing within intermittently connected mobile ad hoc networks," in *Proc. of the 2006 International Conference on Wireless Communications and Mobile Computing,* 2006, pp. 551-556.

[20] K. A. Harras, K. C. Almeroth and E. M. Belding-Royer, "Delay tolerant mobile networks (dtmns): Controlled flooding in sparse mobile networks," *Lecture Notes in Computer Science,* vol. 3462, 2005, pp. 1180-1192.

[21] D. Henriksson, T. F. Abdelzaher and R. K. Ganti, "A caching-based approach to routing in delay-tolerant networks," in *Computer Communications and Networks (ICCCN), 2007,* pp. 69-74.

[22] S. Jain, K. Fall and R. Patra, "Routing in a delay tolerant network," *ACM SIGCOMM Computer Communication Review,* vol. 34, 2004 pp. 145-158.

[23] E. Jones, L. Li, J. Schmidtke and P. Ward, "Practical routing in delay-tolerant networks," *IEEE Transactions on Mobile Computing,* vol. 6, 2007, pp. 943-959.

[24] E. Jones and P. A.S. Ward. "Routing strategies for delay-tolerant networks," *Submitted to Computer Communication Review CCR*, 2006.

[25] H. Kang and D. Kim, "Vector Routing for Delay Tolerant Networks," *IEEE Vehicular Technology Conference*, Calgary, B.C., Canada, pp. 1–5, September 2008.

[26] H. Kang and D. Kim, "HVR: History-Based Vector Routing For Delay Tolerant Networks," in *Proc. of the IEEE International Conference on Computer Communications and Networks*, San Francisco, C.A., United States, pp. 1–6, August 2009.

[27] A. Krifa, C. Barakat and T. Spyropoulos, "Optimal buffer management policies for delay tolerant networks," in *5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON'08),* 2008, pp. 260-268.

[28] Y. B. Ko and N. Vaidy, "Location-Aided Routing in Mobile Ad-Hoc Networks," *ACM Wireless Networks Journal*, June 2000.

[29] J. F. Kurose and K. W. Ross, “Computer Networking: A Top-Down Approach, Fifth Edition,” Pearson Education INC., 501 Boylston Street, #900, Boston, MA 02116, 2010.

[30] Q. Li and D. Rus. “Communication in Disconnected Ad Hoc Networks Using Message Relay,” *Journal of Parallel and Distributed Computing.*, 63(1):75–86, 2003.

[31] A. Lindgren, “Evaluation of Queueing Policies and Forwarding Strategies for Routing in Intermittently Connected Networks,” *First International Conference on Communication System Software and Middleware (COMSWARE),* 2006*,* pp. 1-10.

[32] A. Lindgren, A. Doria and O. Schelen, "Probabilistic routing in intermittently connected networks," *Lecture Notes in Computer Science,* 2004*,* pp. 239-254.

[33] C. Perkins, E. Belding-Royer and S. Das, “Ad-Hoc On-Demand Distance Vector (AODV) Routing,” RFC 3561, IETF Network Working Group, July 2003.

[34] P. Ranjan, P. K. Saraswat, A. Kumar, S. Polana, and A. Singh, “wildCENSE - sensor network for wildlife monitoring,” Technical report, Dhirubhai Ambani Institute of Information and Communication Technology, May 2006.

[35] J. Rice, “Seaweb Acoustic Communication and Navigation Networks”. in *Proc. of International Conference Underwater Acoustic Measurements: Technologies & Results,* July 2005.

[36] M. Sede, X. Li. "BLER: Routing in Large-Scale Buses Ad Hoc Networks," IEEE Wireless Communication and Networking Conference , WCNC’08, March, 2008.

[37] R. Shah, S. Roy, S. Jain, W. Brunette, “Data MULEs: Modeling a Three-tier Architecture for Sparse Sensor Networks”, *IEEE SNPA Workshop*, May 2003

[38] T. Small and Z. J. Haas, “The Shared Wireless Infostation Model: A New Ad Hoc Networking Paradigm (or Where there is a Whale, there is a Way),” in *Proc. of MobiHOC*, pp. 233– 244, 2003

[39] T. Spyropoulos, K. Psounis and C. S. Raghavendra, "Spray and wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks," in *Proc. of the 2005 ACM SIGCOMM Workshop on Delay-Tolerant Networking,* 2005, pp. 252-259.

[40] T. Spyropoulos, K. Psounis and C. S. Raghavendra, "Single-copy routing in intermittently connected mobile networks," in *First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, IEEE SECON* 2004*,* pp. 235-244.

[41] A. Vahdat and D. Becker, “Epidemic Routing for Partially Connected Ad Hoc Networks,” Technical Report CS-200006, Department of Computer Science, Duke University, Durham, NC, 2000.

[42] Y. Wang, S. Jain, M. Martonosi and K. Fall, “Erasure-Coding Based Routing for Opportunistic Networks,” *ACM SIGCOMM Workshop on DTN*, 2005

[43] F. Warthman: “Delay-Tolerant Networks (DTNs) - A Tutorial”, Warthman Associates, 2003; http://www.dtnrg.org/docs/tutorials/warthman-1.1.pdf

[44] J. Widmer and J.-Y. LeBoudec, “Network Coding for Efficient Communication in Extreme Networks,” in *WDTN*, Aug. 2005.

[45] The Wizzy Project: http://www.wizzy.org.za/

[46] J. Wu, S. Yang, and F. Dai, “Logarithmic Store-carry-forward Routing in Mobile Ad Hoc Networks,” *IEEE Transactions on Parallel and Distributed Systems*, 18(6), June 2007.

[47] P. Zhang, C. M. Sadler, S. A. Lyon and M. Martonosi, "Hardware design experiences in ZebraNet," in *Proc. of the 2nd International Conference on Embedded Networked Sensor Systems,* 2004, pp. 227-238.

[48] Z. Zhang, “Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges,” in *IEEE Communications Surveys & Tutorials,* vol. 8, 2006, pp. 24-37

[49] Z. Zhang and Q. Zhang, “Delay/Disruption Tolerant Mobile Ad Hoc Networks: Latest Developments,” Wiley InterScience, *Wireless Communications and Mobile Computing*, pp. 1219–1232, 2007

[50] W. Zhao, M. H. Ammar and E. Zegura, “A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks,” in *Proc. of ACM MobiHoc*, pp. 187–198, 2004.

[51] W. Zhao, M. H. Ammar and E. Zegura, “Controlling the Mobility of Multiple Data Transport Ferries in a Delay-Tolerant Network,” *24th Conference of the IEEE Computer and Communications Societies (IINFOCOM),* Miami, FL, USA., 2005.

[52] W. Zhao, M. Ammar and E. Zegura, "Multicasting in delay tolerant networks: Semantic models and routing algorithms," in *Proc. of the 2005 ACM SIGCOMM Workshop on Delay-Tolerant Networking*, 2005, pp. 268-275.

[53] A. Krifa, C. Barakat and T. Spyropoulos, “An Optimal Joint Scheduling and Drop Policy for Delay Tolerant Networks,” *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, Workshop on Autonomic and Opportunistic Communications*, Newport Beach, C.A, United States, June 2008.

[54] M. Prathaban and J. Kohlenberg, “Buffer Management Policy for Mars Intelligent Proximity Network,” Proceedings of the IEEE Second Asia International Conference on Modelling and Simulation, May 2008.

[55] M. C. Chuah and W. B. Ma, “Integrated Buffer and Route Management in a DTN with Message Ferry,” Journal of Information Science and Engineering, Vol. 23, pp. 1123–1139, February 2007

[56] Y. Li, L. Zhao, Z. Liu and Q. Liu, “N-Drop: Congestion Control Strategy under Epidemic Routing in DTN,” Proceedings of the ACM International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly, Leipzig, Germany, pp. 457–460, June 2009.

[57] D. Wick, “Delay Tolerant Networks in a Nutshell,” Bachelor Thesis, University of Bern, August 2007.