Disruption-Tolerant Networking: A Comprehensive Survey on Recent Developments and Persisting Challenges

Maurice J. Khabbaz, Chadi M. Assi, and Wissam F. Fawaz

***Abstract*—Nowadays, wireless networks are witnessing several deployments in various extreme environments where they suffer from different levels of link disruptions depending on the severity of the operating conditions. In all cases, their operation requirements are differently altered and their performance is negatively affected rendering them heterogeneous by nature. In the open literature, these networks are known as Intermittently Connected Networks (ICNs). The existing Internet protocols fail to operate properly in the context of ICNs, thus raising a variety of new challenging problems that are attracting the attention of the networking research community. Delay-/Disruption-Tolerant Networking emerged as a highly active area of research where networking experts compete in addressing the various ICN problems. Over time, unicast routing, one of the architectural key components common to all ICNs, became an almost independent field of research in which significant efforts continue to be invested. In contrast, network architectural designs, scheduling and forwarding issues dating from the early days of Inter-Planetary Networks (IPNs) have received relatively little attention and accumulate numerous pending challenges. Moreover, the gap caused by the lack of accurate ICN mathematical models is still large irrespective of some of the appreciated seminal works in this direction. This paper sheds the light over the latest advancements in each of the above-mentioned research sectors and highlight pending open issues in each of them.**

***Index Terms*—Delay-tolerant, bundle protocol, routing, ICN, cooperative DTNs, DTN architecture.**

I. Introduction

TRADITIONALLY, data networks are modeled using connected graphs whereby the existence of at least one end-to-end path between any source-destination pair is always guaranteed. In these networks, any arbitrary link connecting two network nodes is assumed to be bidirectional supporting symmetric data rates with low error probability and latency ($i.e.$ round-trip time is in the order of milliseconds). Additionally, network nodes suffer infrequent power outages and thus remain functional most of the time. Incoming packets are buffered until they are further forwarded to their respective next hops (in case the current node is an intermediate node) or successfully received and processed by their intended receiving application (in case the current node is an ultimate receiver/destination). In this context, packets are not supposed to reside in a node's buffer for a long period of time. Based on this assumption, buffer sizes are relatively small and optimized in such a way to keep a low overall packet drop rate due to buffer overload.

Following these fundamental assumptions, the Internet, the global packet switching network, was conceived and its operating protocols, particularly the TCP/IP protocol suite, were developed. However, such assumptions may not be appropriate when modeling existing and recently emerging wireless networks, especially those deployed in extreme environments ($e.g.$ battlefields, volcanic regions, deep oceans, deep space, developing regions, etc.) where they suffer challenging conditions ($e.g.$ military wars and conflicts, terrorist attacks, earthquakes, volcanic eruptions, floods, storms, hurricanes, severe electromagnetic interferences, congested usage, etc.) resulting in excessive delays, severe bandwidth restrictions, remarkable node mobility, frequent power outages and recurring communication obstructions. Under such conditions, wireless network connectivity becomes considerably intermittent and the existence of contemporaneous end-to-end path(s) between any source-destination pair can no longer be guaranteed. Unusual and repetitive occurrences of network partitioning and topology changes occur; thus, it is utterly probable that two nodes currently co-existing in an arbitrary connected portion of the network may not co-exist in that same or any different connected network portion in the future. Due to low power, network nodes often unexpectedly shut down or enter sleep mode for energy saving resulting in frequent link disruptions. Data transmission rates become highly asymmetric and links highly error prone. When coupled with relatively small node buffer sizes, buffer overload becomes a severely penalizing problem as it exponentially increases the packet drop rate.

Popular examples of such intermittently connected networks (ICNs) scenarios, which have already been the subject of extensive research, include:

1) Exotic Media Networks (EMNs) interconnecting extra-terrestrial nodes ($e.g.$ satellites, deep space probes, landers, orbiters, etc.) that may also periodically communicate with ground ($i.e.$ terrestrial) nodes using high latency Radio Frequency (RF) transceivers such as those used in the Inter-Planetary Network (IPN) project.
2) Wireless Sensor Networks (WSNs), Mobile Wireless Sensor Networks (MWSNs) and Sensor/Actuator Networks (SANs) deployed in extreme regions ($e.g.$ Amazons, deep volcanic or underwater areas, etc.), which

Manuscript received 14 July 2010; revised 18 January 2011 and 13 April 2011.

M. Khabbaz and C. Assi are with Concordia University (e-mail: {mkhabbaz, assi}@ece.concordia.ca).

W. Fawaz is with the Lebanese American University (e-mail: wissam.fawaz@lau.edu.lb).

Digital Object Identifier 10.1109/SURV.2011.041911.00093

consist of low power sensor nodes that, for the purpose of energy saving, periodically switch between active/sleep modes and thus are unable to continuously communicate with a data-collecting server.

3) Mobile Ad-Hoc Networks (MANETs) typically consisting of nodes ($e.g.$ GPSs, PDAs, Cellular Phones, Tracking devices, Laptops, etc.) mounted over continuously moving objects ($e.g.$ vehicles, moving individuals, animals, etc.). Communication in such networks is frequently interrupted either due to nodes going out of communication range of each other or obstructing obstacles or node destructions as is the case in Battlefield Wireless Military Networks (BWMNs).

The ICN heterogeneity and mutual independence is, at this point, quite obvious. Each ICN imposes different specific requirements for proper operation and only adapts to a very limited communication region with relatively homogeneous characteristics, as illustrated in Figure 1. At this stage, it is absolutely legitimate and reasonable to declare the fated failure of the adaptation attempt of the majority of the Internet protocols to such challenged networks. This is especially true since challenged networks violate one or more of the previously mentioned major characteristics and assumptions based on which such protocols were developed.

As their popularity increased over time, there was an urgent need for information exchange between ICNs, particularly Mobile Wireless Sensor Networks (MWSNs) and the Internet, [2], [3]. For sole interconnection purposes with the Internet, special types of nodes were engineered and integrated within each ICN. The term *gateways* was employed in the open literature to refer to such nodes, since each of them implements an ICN-customized application layer gateway aiming at translating Internet protocol parameters to their ICN-specific counterparts. It is also possible that gateways, in some cases, be equipped with buffers used for holding messages until delivered to their intended next hop. However, the haphazard design and implementation of such gateways in an unstructured ICN-specific way resulted in severely limited ICN interoperability, [2].

Now, regardless of all those stringent limitations imposed by ICNs and their extreme environments, a wide variety of wireless applications could still be supported. Dissemination of location-dependent information ($e.g.$ traffic reports, available parking lots, etc.) through Vehicular Ad-Hoc Networks (VANETs), [4], low-cost Internet supply to isolated villages and developing communities, [5], underwater acoustic networks, [6], Pocket-Switched Wireless Networks (PSWN) ($e.g.$ PeopleNet) used as an extension to Internet access point connectivity, [8] and so forth, all are legitimate examples of such applications. Nonetheless, enabling the proper operation and functionality of such applications under these challenging conditions pushed researchers to propose a new networking paradigm referred to in the open literature as a *Delay-* or *Disruption-Tolerant Networking*.

Over the last decade, Delay-/Disruption-Tolerant Networking has been a highly active area of research. Networking architecture and application designs, routing, multicasting, delay and buffer management, congestion and flow control, cooperative schemes and mathematical modeling, all persist as very hot topics attracting the attention of a large body of researchers. Some seminal works and significant contributions have been made in those sectors. The Delay-Tolerant Networking Research Group (DTNRG) [9] not only embodies huge archives of valuable published papers and past discussions but also points to very recent updates in the field. Two existing surveys, [10] and [11], provide excellent background on developments in particular related areas that took place within a limited epoch before mid 2007. Ever since, the large amount of continuously emerging new studies and solution proposals to the various ICN problems, clearly indicates the networking community's surge of interest in this important topic. However, to the best of our knowledge, recent works have not yet been clearly summarized in a well organized manuscript that complements the earlier referenced surveys. In addition, broad and further in-depth studies are still required before large-scale ICN deployments prevail. Consequently, the main contributions of this manuscript are three-fold:

1) Present a comprehensive tutorial-like study that exposes the Delay-/Disruption-Tolerant Networking problem at large.
2) Briefly summarize the previously published surveys and supplement them by reviewing a wide scope of recent works and solution proposals that span the whole ICN research field.
3) Present a list of the major persisting open research issues.

The rest of this manuscript is organized as follows. Section II discusses the basic DTN concepts. In section III we survey DTN architectural designs. Section IV present a general overview of the DTN's Bundle Protocol. Section V surveys the recent DTN forwarding protocols. Section VI discusses the recent works on cooperative strategies in DTNs. Section VII presents important open research problems. Concluding remarks are provided in section VII. Finally, section IX lists the top must read references.

II. Delay-/Disruption-Tolerant Networking

A. Concrete Definition

The Inter-Planetary Network (IPN) [12], a project launched in 1998, aimed at establishing connectivity between nodes arbitrarily located on the different solar system planets. Nevertheless, deep-space communication was subject to excessive propagation delays, high error rates and frequent disruptions causing the failure of the traditional protocols used to transmit data packets over terrestrial networks. As such, protocols used for deep-space communication have to be delay and disruption-tolerant. The networking research community then realized that IPN is a special scenario of a broader class of challenged networks, known also as Intermittently Connected Networks (ICNs), that may either be integrated into the terrestrial Internet or simply deployed at its edges, as illustrated in Figure 2.

In the wide variety of work published over the past decade ($e.g.$ [14]) researchers identified the different communication disruption and dealy causes in various heterogeneous challenged networks ($e.g.$ EMNs, IPNs, MWSNs, SANs, MANETs, VANETs, BWMNs, etc.) and

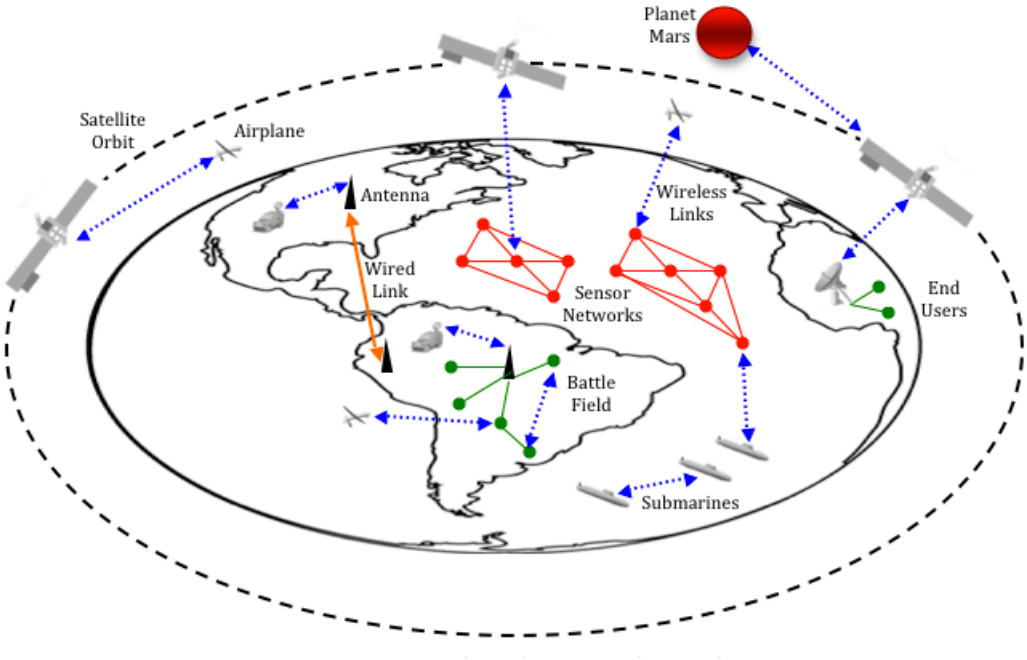

Fig. 1. Independent Regional Networks.

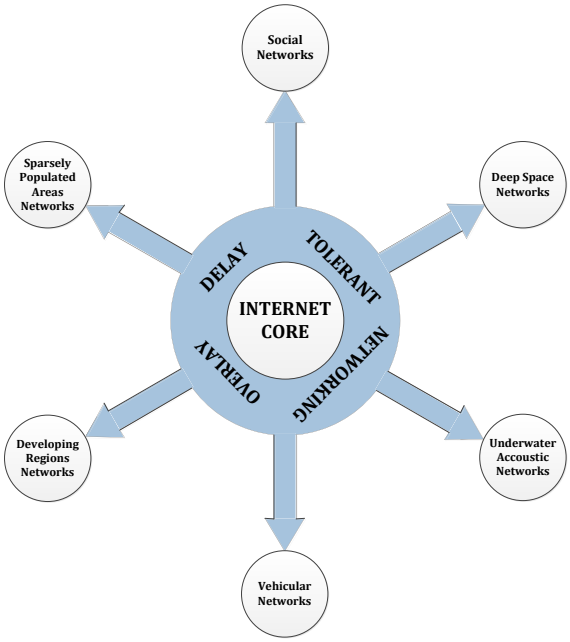

Fig. 2. Delay-/Disruption-Tolerant Networking Overlay.

thoroughly described the characteristics of those networks and the requirements to achieve interoperability between/among them. In addition, through various philosophies, researchers competed in concretizing a definition of this novel networking paradigm. Those definitions were carefully supported by several examples extracted from real-life scenarios. We observed that the terms *Intermittently Connected Networks* (ICNs), *Challenged Networks* and *Delay-/Disruption-Tolerant Networks* were used interchangeably to refer to the network itself. It is however important to note in this regard that *Delay-/Disruption-Tolerant Networking* is an architecture proposed by the Internet Engineering Task Force (IETF) to handle ICNs, [15]. Hence, Delay-/Disruption-Tolerant Networking and ICN are two totally different concepts. It is therefore worthwhile to concretely define them.

1) Intermittently Connected Network:

An Intermittently Connected Network, also known as a Challenged Network, is an infrastructure-less wireless network that supports the proper functionality of one or several wireless applications operating in stressful environments, where excessive delays and unguaranteed continuous existence of end-to-end path(s) between any arbitrary source-destination pair, result from highly repetitive link disruptions.

The above definition implies the presence of an elevated level of inherent uncertainty in such networks where any arbitrary node totally lacks network state information ($i.e.$ information about other nodes in the network, network topology, etc.) and thus has to devise its own operating decisions. The effect of this uncertainty is twofold: First, at an intermediate stage, messages arrive to nodes with unknown next hops and it is often the case where such nodes are even obliged to accumulate and queue arriving messages long enough until an appropriate next hop becomes available. Typical small buffer sizes are therefore obviously unsuitable for such conditions. Instead, to achieve data delivery, ICN nodes are augmented with permanent storage capabilities and equipped with relatively large buffer sizes enabling them to indefinitely carry messages until they can be further forwarded, a technique known in the open literature as *store-carry-forward*. Second, unknown network information impels nodes to disseminate multiple message copies to other nodes in an attempt to increase the message delivery probability, a technique referred

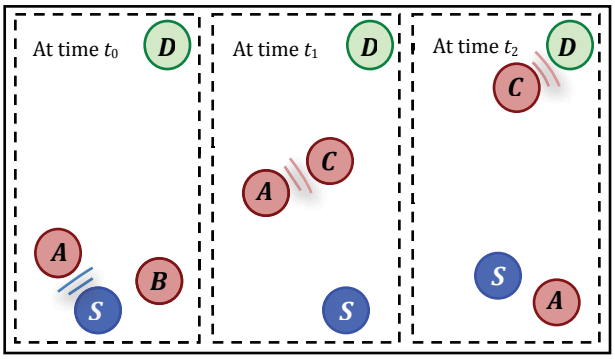

Fig. 3. Illustrative example of a time evolving DTN.

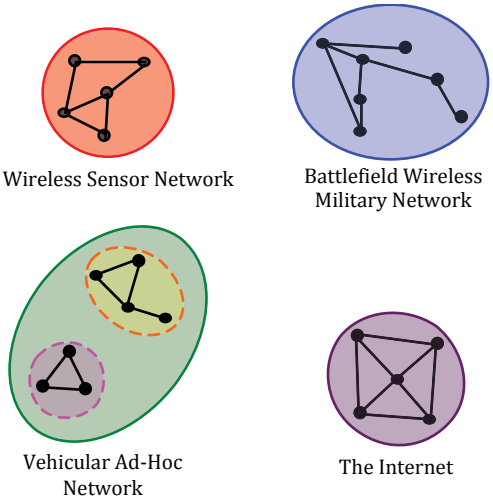

Fig. 4. Disconnected Regional Networks.

to as *flooding*. Flooding the network haphazardly causes a relatively rapid nodal buffer overflow and therefore an increase in the drop rate. Hence, buffer space emerges as a crucial resource to be effectively managed. Figure 3 illustrates an example of a time evolving ICN where a direct end-to-end path between a source node S and a destination node D never exists at any point in time.

Successful delivery of messages from S to D can be achieved only if intermediate nodes receive those messages from S and carry them to D. At time t_0, the source S generates a message addressed to D. However, node D is unreachable neither through direct contact nor through any pre-determined end-to-end path. Instead, S finds itself able to communicate with either node A or B, both happening to be in its communication range. Based on some information, S decides to forward the message to node A that in turn stores the message in its buffer hoping to be able to either directly deliver it to its ultimate destination, D, or forward it to any other intermediate node. At time $t_1(t_1 > t_0)$, A enters in direct communication with node C to which it decides to forward S's message hoping that C will be able to successfully deliver it to D. C receives the message and stores it in its buffer. Finally, at time $t_2(t_2 > t_1)$, C enters into direct communication with D and therefore successfully delivers the message from S.

2) Delay-Tolerant Networking:

Delay-/Disruption-Tolerant Networking is an overlay architecture intended to operate above the protocol stacks of the distinct ICNs and enable gateway functionality between them through the use of storage capacity, a variety of protocol techniques, replication and parallel forwarding, forward error correction and many other techniques for overcoming communication impairments.

In Figure 3, we took a closer look at how ICN nodes behave. However, it is always a good idea to zoom out a little and take a global view especially when it comes to understanding the Delay-/Disruption-Tolerant Networking architecture. For illustrative purposes, Figure 4 shows four independent and disconnected heterogeneous regional networks. Such regional networks are best described as being *isolated* from each other. The DTN architecture, [15], targets at interconnecting those regional networks as well as setting up a new framework to handle their heterogeneity. Although IP accounts for such heterogeneity, however, it only does so under the restrictive assumptions that we mentioned. To alleviate those assumptions, the DTN architecture supports new addressing schemes and semiotics, a new Bundle Protocol layer extending the traditional IP encapsulation process and finally persistent storage capabilities to survive long period disruptions and system restarts. In view of this, the DTN architecture appears as an overlay layer *transiently* connecting isolated heterogeneous networks together thus ensuring interoperability between them irrespective of their underlying technologies, protocol layers and region specific characteristics.

Figure 5 conceptually demonstrates the previously explained concept.Networks in each of the isolated regions may be either well connected or intermittently connected. An intermittently connected regional network would be a very special case as it can be considered as two separate regions using the same underlying technology ($e.g.$ in a battle field a node connecting two other portions of battlefield network may be hit and destroyed leaving behind it a partitioned network). Nodes from any region may disengage and transfer to any other region. Links to any transferring node get disrupted as soon as this node gets out of communication range of other nodes in the network ($e.g.$ node A on its way from region (1) to region (2) and node B that has just left region (2) and is transferring to region (3)). However, a node that is about to enter a certain region ($e.g.$ node C entering region (3)) starts broadcasting its presence so as to establish communication with any nearby node that happens to be in range. Links between the two nodes gets established and the new incoming node becomes a member of the regional network it is visiting.

3) Naming Convention: The term *Delay-/Disruption-Tolerant Network* (DTN) has been widely adopted in several works ($e.g.$ [11], [55], [70], [77], [88] through [92], [94], [95], [100], [107] through [113] and [116]) as a reference to a network that is subject to repetitive link disruptions and

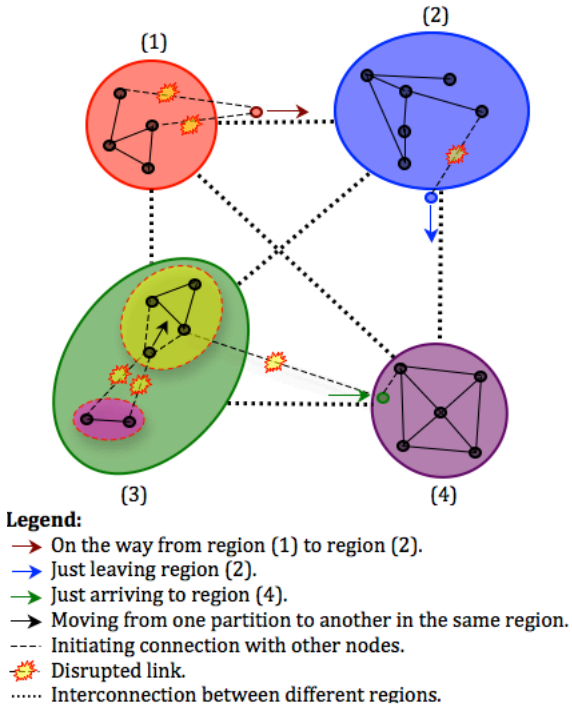

Fig. 5. Targeted interconnection of regional networks through an overlay DTN architecture.

excessive delays, in other words an Intermittently Connected Network (ICN). As we pointed out earlier, *DTN* stands for Delay-/Disruption-Tolerant Networking which is an overlay architecture designed to handle ICNs. In light of the above, we suggest to use DTN in reference to the network by itself and Delay-/Disruption-Tolerant Networking in reference to the DTN architecture. We use this naming convention in the rest of this manuscript.

B. DTN Problem Generalization

DTN application scenarios are manifold. Nonetheless, a persisting challenge is the development of a *single* protocol stack that is able to handle this huge diversity of DTN applications. Existing Internet routing protocols are all based on fundamental assumptions, particularly: a) relentless connectivity and b) minimal delays where instantaneous responses take place through closed control loops. In contrast, DTNs operate in extreme environments under highly challenging conditions where frequent connectivity intermittence becomes the norm rather than the exception causing severe network partitioning, excessive latencies, high packet loss and bit error rates, reduced and assymetric bandwidth. Hence, such networks are simply uncontrollable using closed control loops and contemporaneous end-to-end paths between any arbitrary source-destination pair cannot be guaranteed. Instead, each network node must operate independently as all others are often unavailable to cooperate. Under such conditions, timely data delivery become almost far fetched.

Furthermore, transmission reliability is still debatable. While some applications ($e.g.$ online banking, telecommand, E-mail, critical file transfers, etc.) require complete reliability or in other words 100% delivery ratio, others such as traffic management and statistical telemetry ($e.g.$ monitoring applications for pollution, weather conditions, virus spreading, etc. where information data is collected periodically within tight time intervals) do not have such stringent constraints. Ensuring 100% delivery ratio for non-critical reliability applications may be exhaustive and will severly degrade the overall system's performance. Imagine that, at some point in time, an arbitrary node's buffer was loaded with non-critical messages and suddenly a critical condition occurs. Thus, all arriving critical messages will either be trapped behind non-critical ones or simply dropped due to buffer overload. In other words, valuable storage resources are being quite inefficiently utilized to ensure 100% reliability to non-critical messages. Second, the required multiple transmissions that ensure timely delivery will rapidly deplete the node's battery and cause it to shut down and completely disappear from the network. In both cases the sensor node will become useless.

E-mail is an example of an application that requires full reliability but is more tolerant in terms of delivery latencies. In contrast, some web-based applications require stringent delivery timliness where otherwise data becomes completely useless. Other applications such as real-time telecommand has strict requirements in terms of both high delivery ratio and low delivery latencies. At this stage, one may reasonably question whether delay minimization contradicts the concept of delay-tolerance in the DTN context. Actually it does not. As a matter of fact, one DTN application can be designed to tolerate delays in the oder of a couple of minutes while another application may tolerate delays in the order of days or possibly one or several weeks. As such, in light of the above discussion, service differentiation becomes obviously necessary and is closely dependent on the DTN-application itself and the environment under which this application is expected to operate. Hence, the design of a universal reactive protocol stack that is able to handle all DTN application scenarios is clearly not a wise approach to follow. Instead, proactive, service-requirements-aware designs emerge as more promising solutions.

C. A Promising Solution Approach

The starting point of an appropriate DTN protocol design is the fact that DTN nodes are mainly mobile wireless devices with resource constraints ($i.e.$ storage capacity and power). Given a certain application with possibly some additional limitations imposed by its operating environment, the protocol designer will be able to determine what requirements the deployed DTN is expected to satisfy. Depending on the deployed application, these requirements are either maximization of delivery ratio or minimization of delivery latencies or possibly both. Numerous works dating from the old days of the Internet and up to some of the most recent protocol designs [2] and [52] through [116], utilized the delivery ratio and delivery delay as metrics to reactively evaluate the performance of the different devised protocols. In contrast, in this manuscript, reliability and latency are rather interpreted as service objectives that a DTN is required to achieve. This is where proactiveness comes into play in the DTN protocol design process.We agree with the authors in [17] who observed that message duplication, on

one hand, increases the delivery ratio and, on the other hand, it decreases the delivery delay. In fact, whenever message replicas are distributed among several nodes, if some of these nodes disappear ($e.g.$ failures, destruction, power outage, alternation between sleep and active modes, etc.) the task of message delivery is delegated to the other remaining active nodes. In addition, as the number of carrier nodes increases, the chance that one of them encounters and delivers the message to its ultimate destination increases. Hence, message deliveries are more likely to become faster. In other words, the delivery delay decreases. Nevertheless, there are two other important observations highlighted in [17], namely: message duplication, first, is power consuming due to multiple message transmissions and receptions, and second, requires more buffer space to store the message replicas. Therefore, DTN nodes become more prone to battery depletion and buffer overloads. Nodes that unexpectedly shut down or suffer low buffer space are simply useless and negatively impact the network's performance. Therefore, in a DTN composed of n heterogenous nodes, we believe that the number of nodes carrying replicas of M can vary between 1 and n depending on both the severity of nodal resource limitation and the required service objective which differs from one application to another.

As far as the traditional Internet is concerned, high reliability and low delivery latency can be jointly achieved. In contrast, both the severity of the operating environment and the heterogeneity of nodal resource constraints and settings significantly limit the capabilities of DTNs. Yet, the envisioned DTN applications have minimum requirements that have to be satisfied for normal operation. To this end, the authors of [16] enumerated the diverse device classes that may participate in a DTN ($e.g.$ laptops, PDAs and other handheld devices, sensors, etc.) and devised an analytical model to evaluate the impact that the differentiated nodal characteristics, resource availabilities, and mobility patterns have on the performance of DTN protocols. This study is of particular appeal and the proposed model sounds quite promising. However, it requires further refinements. Throughout their study, the authors seem to focus only on the delivery delay. The delivery ratio seems to be overlooked. In addition, this proposed model is based on a strictly unrealistic assumption that network nodes are continuously *au courant* of the different network activities. For example, upon successful delivery of a particular message M, all other nodes carrying copies of M are immediately informed to delete these copies from their buffers and free up storage space. While this type of feedback can be realized through properly designed protocols, in the context of DTNs, such feedback is not immediate and may take some time before all carrier nodes are informed of a successful delivery. This surely has a negative impact on the network's performance.

The authors of [17] accounted for both nodal heterogeneity and service objectives while strictly abiding by the general fundamental characteristics of DTNs: a) unavailability of network knowledge, b) storage and power contraints and c) the possibility of the existence of several applications running concurrently over one DTN. Following a long reasoning and a sequence of consensus statements, the authors conjecture that, in the context of DTNs, it is too difficutlt, if not impossible, to achieve both high delivery ratio and low delivery delays simultaneously. As such, they conclude that there exists a trade-off between the two service objectives and propose two important design positions:

- *Position I*: High reliability is achieved by making *judicious* decisions.
- *Position II*: Low delivery delays are achieved by undertaking some risks.

As far as the first position is concerned, wiseness comes from knowing how to conserve valuable network resources for the right time. Intuitively, the distributed copies of a particular message are deleted once the carrier nodes are informed of the corresponding message's successful delivery to its ultimate destination. Accordingly, buffer space is restored. However, multiple message copies are injected into the network through multiple transmissions, each of which consumes a considerable amount of non-renewable power. Greedy message duplication will therefore cause devices to rapidly shut down due to power outage. Thus, all messages stored in dead devices are practically lost resulting in a significant decrease in delivery ratio.

Regarding the second position, the authors of [17] summarize that risk is taken when messages are sent out to nodes with unpredictable delivery probabilities (i.e. probabilities of encounter with the ultimate message destination). They reasonably argue that replicating a message to a relatively large number of nodes increases the probability that one of these carriers encounters the destination within the cutoff time. In the same direction, we also contend that risk is also taken when foolishly replicating messages to nodes that are unable to accomodate the incoming replica due to critical buffer or unavailable buffer space. As such, not only would the transmitted copy not contribute to delay minimization but also would waste valuable transmitter power and contribute to reducing the overall network reliability.

To this end, we conclude that the presence of as low as a single message copy in the network might not be enough to meet a predefined DTN application service objective. However, in light of the above discussion, we contend that neither as many replicas as there are active nodes in the network would also do any good. Again, DTN protocols that account for both nodal resource diversity, networking environment heterogeneity and application service objectives and accordingly perform controlled message replication, seem to be quite promising solutions.

III. DTN Architectural Designs

In this section, we shed the light on DTN architectural design-related issues. We instigate our exploration from the deep-space Inter-Planetary Networking (IPN) architecture, mother of all subsequent DTN architectural developments and extend to cover the most important recently emerging terrestrial DTN prototypes and their pertaining architectural enhancements.

A. The Inter-Planetary Networking Architecture

DTN architectural designs and explorations dated ever since the first Inter-Planetary Internet (IPI) project [12] started. After thorough examination, researchers concluded that the

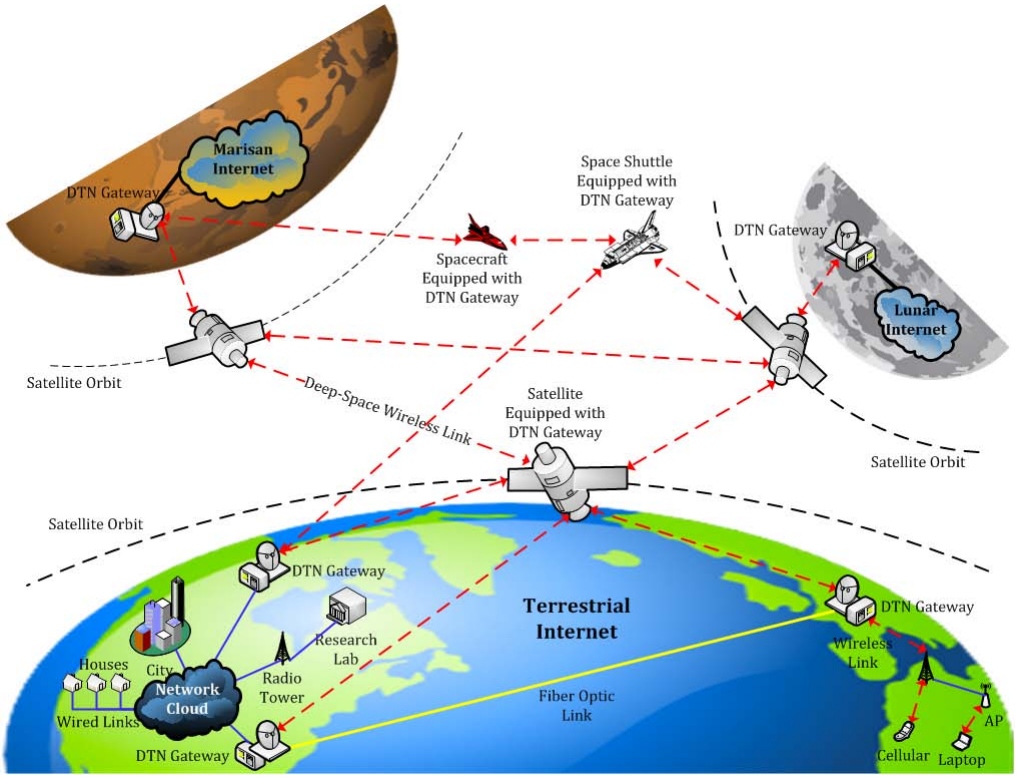

Fig. 6. Our vision of the so pictured IPN at the time the IPN architectural definition was devised by V. Cerf et.al. in 2001.

astronomical mechanics of the solar system and the immutable characteristics of the outer space render end-to-end operation of standard Internet protocols in such environments infeasible. In an attempt to overcome this, they pictured the extra-terrestrial world as the *catholic network* of ordinary Internets. A partition of independent regions, each of which is deploying an ordinary Internet. At the edge of each regional network may be found one or several gateways that are supposed to operate cooperatively over deep-space communication links so as to establish a stable inter-planetary backbone and interconnect all the regions together. To this end, the lion's share of efforts were invested in addressing network inter-operability and interconnection-related issues. Figure 6 illustrates our vision of how the first IPN network was pictured back in 2001.

The author, in [14], presented a thorough investigation of critical DTN protocol design-related issues and proposed the Delay-Tolerant Networking overlay architecture as a comprehensive solution to the DTN inter-operability problem. This seminal study constituted the essence of several ex cathedral documents that followed ($e.g.$ [18]). The majority of the studies performed by the DTNRG that followed, addressed a few problems raised in [14] ($e.g.$ transport and security), however remained confined to the limited frame of deep-space networks, [20], [21], [82], [83] and [84]. Other issues such as error detection, custody transfers, congestion control, buffer management, addressing, fragmentation, naming and binding were left untouched by the DTNRG. At this stage, it is important to attract the reader's attention to the fact that IPNs are often very sparse networks where network information ($i.e.$ link availabilities, paths, node mobility, etc.) is often known *a priori* and notice that almost all the present protocol designs within the DTNRG rely on network knowledge and mainly focus on increasing reliability. In contrast, terrestrial DTN scenarios ($e.g.$ disaster response, vehicular communication, etc.) require minimal delivery delays and are highly dynamic and thus network information is not predictable in advance. As a matter of fact, not long ago, it was declared in [3] that the proposed Delay-Tolerant Networking architecture proposed in [14] is unsuitable for terrestrial DTN deployments since, by design, it is not flexible enough to adapt to such scenarios. We believe that, the key to flexibility in this regard is the re-evaluation of the terrestrial application reliability requirements and the relaxation of the stringent network information availability assumption.

B. Terrestrial DTN Architectures

Delay-/Disruption-Tolerant Networking is a newly emerging field of research. Eventhough the majority of the architectural designs mainly focused on deep-space communications, the networking research community witnessed the realization of appealing prototypes that aim at evaluating the performance of protocols specifically designed for terrestrial environments. In this subsection, we discuss the most important prototypes.

1) Wildlife Tracking: Animal mobility, migrations and inter-species interactions are of significant importance and contribute to the development and advancement of biology-related sciences. Tracing animal social behaviors out in the wilderness has been receiving a lot of attention on both the biology and computer networking levels. Particularly, the ZebraNet project [22] united the efforts of biologists and computer scientists by equiping zebras with sensor neckbands

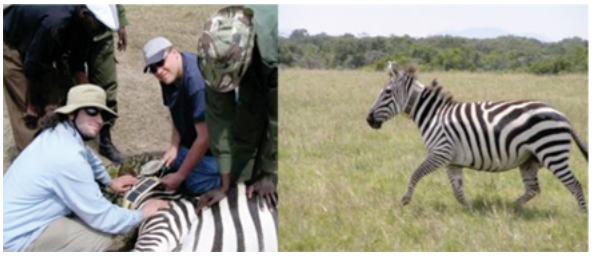

Fig. 7. A zebra being equipped with a sensor collar, [23].

as shown in Figure 7. Using these small sensors, researchers were able to trace animal movements and gather information pertaining to the zebras' behaviors. As such, ZebraNet is an enthralling power-aware wireless ad-hoc sensor network prototype that necessitates momentous bandwidth and more computational power than other sensor networks. It is designed to satisfy the stringent needs of biologists which require real breakthroughs in wireless protocols, low-power system design and power management. More specifically, ZebraNet targets the development, evaluation, implementation, and testing of systems that coalesce computing, wireless communication and permanent storage with Global Positioning Systems (GPS) and other sensors. While the biological observations are outside the scope of this manuscript, reference [22] encloses valuable information regarding energy-memory-successful delivery tradeoffs concluded from testing two DTN routing protocols namely: a) Flooding and b) History-Based Routing.

2) Connectivity in Underdeveloped Regions: Providing connectivity in underdevelopped regions recently received considerable attention. Researchers presented DakNet [24], a project that aims at providing connectivity to remote villages where a few basic connection-enabled computer systems are installed in small booths with access points in order to serve villagers that require access to E-mail, online banking services, governmental services, etc. Requests are all buffered at the access point and are wirelessly opportunistically released to any connection-enabled vehicle ($i.e.$ busses, cars, bikes, etc, that are all equiped with Wi-Fi devices) passing by. In turn, these vehicles will transfer the villagers' requests to the nearest city where they are released over the Internet. Figure 8 illustrates our imagination of DakNet. In this context, it is clear that such connectivity is made possible through primitive, low cost Wi-Fi devices with no infrastructures. As a matter of fact, the major challenge to DakNet is maintaining low setup and operational costs. The authors in [25] exhibited an early thinking in this regard. Following the same approach of [14], they highlight several issues related to routing, security, mobility, location management, naming, addressing and application support. The authors finally informally concluded that, in such networking scenarios, complete reliability and wise data forwarding are quite challenging tasks.

3) Inter-Vehicular and Vehicle-Infrastructure Connectivity: Vehicular networks emerged as a means to enhance traffic safety and reduce the disasterous costs of vehicle collisions. Recently, vehicular communication has received particular attention ($e.g.$ [26] through [32]) in disseminating location-

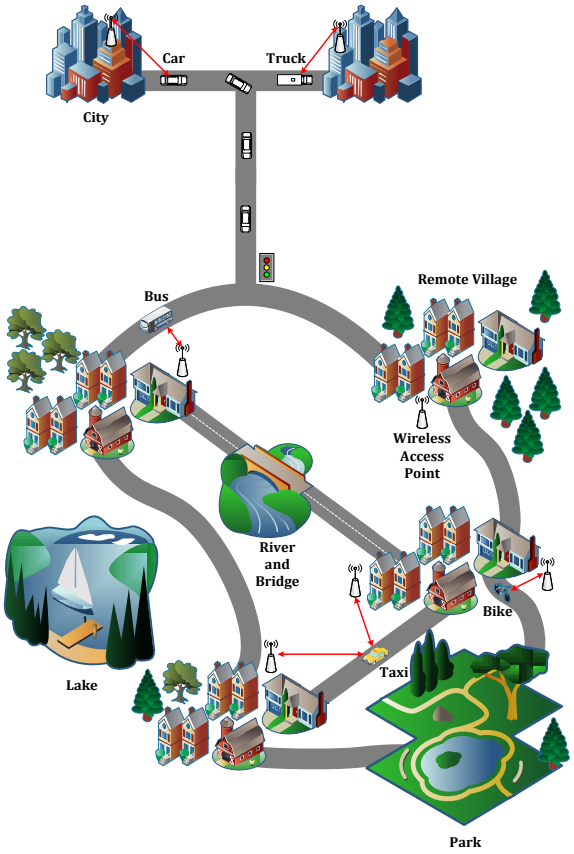

Fig. 8. Our imagination of DakNet.

dependent information ($e.g.$ traffic congestion, availability of parking lots, etc.) as well as providing basic connectivity to passengers travelling via the various conveyance vehicles. As illustrated in Figure 9, vehicular networks are composed of two node types, namely: stationary and mobile. Stationary Relay Stations (SRSs) are deployed along roads and highways. Very few such SRSs, called $gateways$, are privileged by a connection to the Internet or a certain backbone network through minimal infrastructure. All others are isolated and often way apart that they cannot directly communicate. Instead, mobile nodes mounted over vehicles restricted to navigable roadways serve as opportunistic *store-carry-forward* devices that connect any arbitrary SRS pair. Obviously, in such scenarios, delay and disruption-tolerance is a necessity. However, it is worthwhile to note that connectivity patterns differ from one vehicle type to another ($e.g.$ bus, car, truck, metro, train, plane, boat, etc.). Thus, it may sometimes be beneficial to equip large transporters with connectivity infrastructure. As such, some Quality of Service (QoS) requirements may be also met, [33].

4) Social Awareness and Pocket-Switched-based Networking: Social networking is a trend that is recently occupying a large portion of an individual's daily life. Nowadays, almost everybody has a Facebook, Twitter, Windows Messenger, LinkedIn and many other types of profiles through which people of common interests ($e.g.$ friends, relatives, couples, etc.) establish an online presence to socialize and interact with each other and exchange information on so many different levels ($e.g.$ news, event updates, chats, messages, etc.). On-

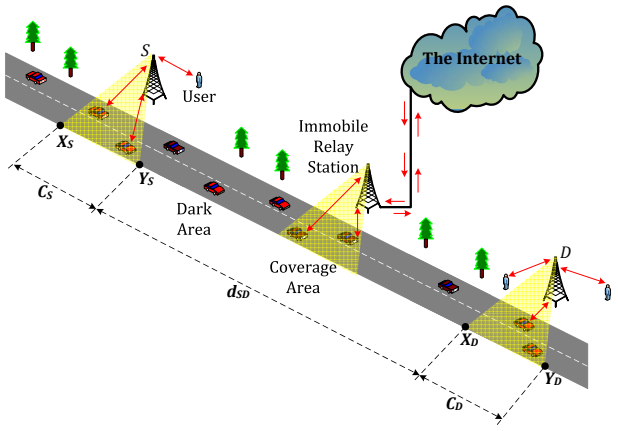

Fig. 9. Vehicular Network.

the-Fly communication or, in other words, communication that requires no Internet connectivity recently emerged as one of the most interesting topics where significant efforts are being invested. Upon opportunistic encounters, wireless handheld devices ($e.g.$ phones, PDAs, laptops, organizers, etc.) may silently interact and possibly exchange information of interest and alert users through pop-up notification messages or beeps. In addition, these mobile devices may also connect to the Internet whenever they fall within the range of a wireless access point. Such communication networks present a number of challenges such as the determination of the amount of information to exchange, the frequency that in-range devices query each other for updates, which devices are chosen among others for information exchange, so on and so forth. PeopleNet [7], Haggle [34], Million-People [35], SocialNets [36] and PeerSoN [37] are examples of top projects that address these different concerns.

IV. The Bundle Protocol

A. Motivation

In the context of the previously discussed deep-space networks (see Figure 6), communication between co-located end points can be perfectly established via typical Internet protocols. The only problem that remains is how to transport data from one region to another through those putative gateways. In view of the limitations of Internet protocols, nothing much can be done unless new protocols, able accomplish this task, are developed. This is actually the first bifurcation from the traditional Internet. New protocols must first account for the limiting factors of the extra-terrestrial world (particularly the variable delays and the haphazard asymmetric bandwidth) and second they must be non-conversational (due to the large BDP values as argued earlier). We conclude that all Internet protocols are to be terminated at IPN gateways and data should be re-encapsulated in a suitable way by the new protocols and travelling over the wireless deep-space links.

Bandwidth mismatch between terrestrial and extra-terrestrial networks also appears as a significant problem. In fact, the best-case sustainable throughput of a TCP connection from Earth to Mars ranges between 1600 bps and 250 Kbps, [12], while terrestrial networks operate over links that are way faster with rates ranging from a couple Mbps to Gbps. In addition, NASA's Deep Space Network (DSN) provides communication resources to most deep-space missions including non-NASA ones. In view of the continuous exponential increase in the number of such missions utilizing those resources, even with the fastest resource upgrades, the DSN's resources will remain over utilized. This adds to the immutable propagation delays some additional overhead emerging from queuing and scheduling delays imposed by terrestrial-resources. Arriving data or data that needs retransmission due to a previous failure may have to be queued until the consecutive contact becomes available ($i.e.$ a period that may be in the order of hours, days or even weeks). Bandwidth mismatch together with delay variability significantly complicate the design of transmission/retransmission timers. This is yet another bifurcation from the traditional Internet. IPN Communication has to be tracked and users need to be informed of errors or delays. But this again imposes a need for interactiveness that was initially ruled out in order to conserve precious bandwidth.

The discussion above adopts IPNs as an illustrative example but it also applies, in a way or another, to general DTNs. It clearly shows that at DTN gateways, a significant deviation from the traditional Internet protocols and mechanisms will take place. However, given the great success of Internet Protocols, the only option for researchers was an attempt to enhance/modify available Internet Protocols to adapt to the new imposed challenges. The question was:

Are there any already well-established mechanisms/protocols used in traditional Internet whose operation over terrestrial networks resembles, even if slightly, to the one required for IPNs? If yes, could those be further enhanced to better fulfill the new challenging requirements of deep-space communication and establish a stable IPN backbone?

After thorough examination of Internet Protocols [12], [14], [15], it was found that electronic mail (E-Mail) provides a nice set of useful features:

1) Asynchronous message delivery.
2) Flexible addressing.
3) Operation over a large set of network technologies.
4) In-Band error reporting.

Nevertheless, E-Mail still fails to operate in challenging environments due to:

1) Utilization protocols that operate on top of TCP ($e.g.$ SMTP).
2) SMTP is a heavy conversational protocol.
3) Lack of dynamic routing.
4) Weak delivery semantics ($e.g.$ limited mail delivery alternatives).
5) Upon delivery failure message is returned to original sender.

While points 1 - 4 seem quite legitimate, one would ask why point 5 would be an issue. Logically speaking, and following a typical real-life postal service scenario, an undeliverable mail is returned to its sender after processing at the post office. However, in a DTN, as explained in [14] and [15], a node, for some reason ($e.g.$ power shortage, buffer exhaustion or

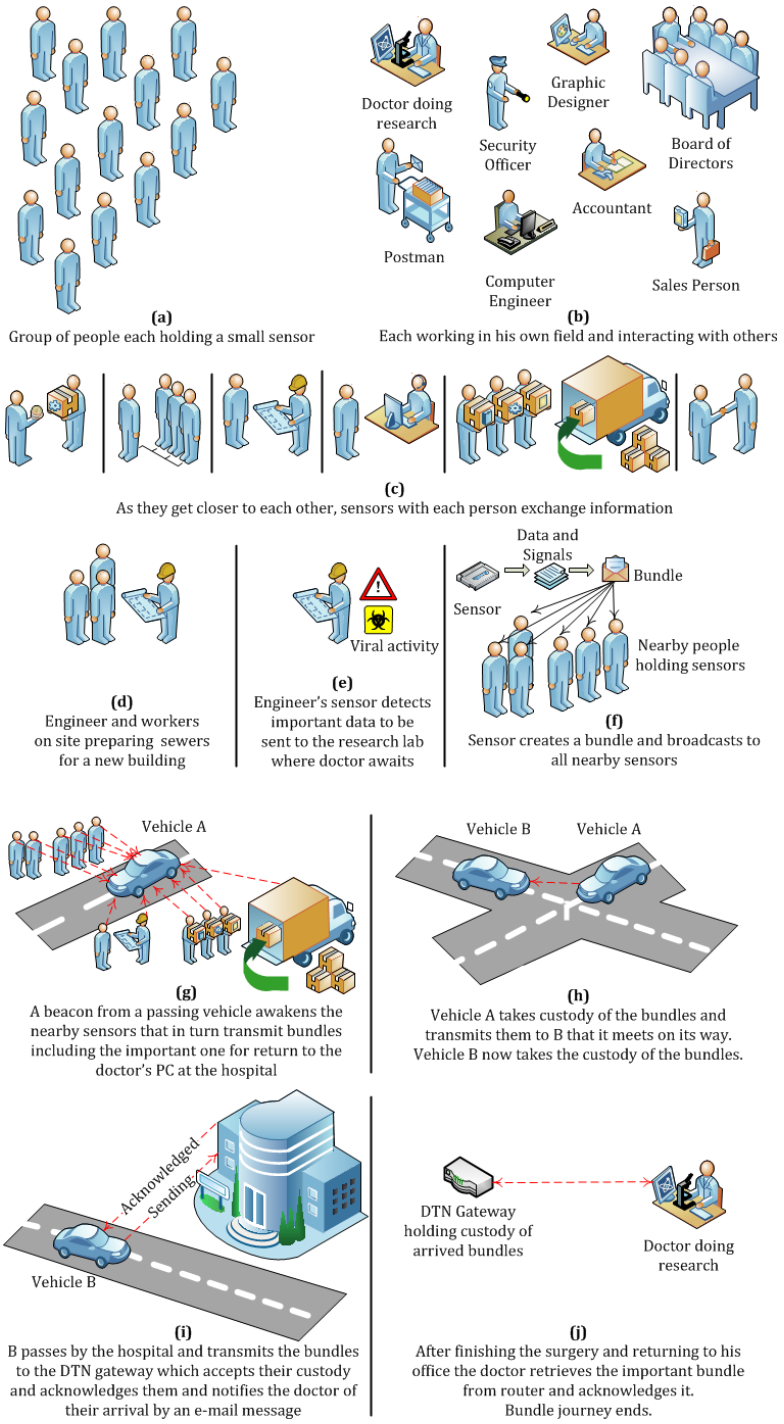

Fig. 10. The journey of a bundle from its source to its destination.

simply has a higher priority task to perform) cannot handle a particular incoming message. In this case, message hand-off to another nearby node can be a nice and useful option.

In conclusion, it is highly likely that an enhanced/modified non-interactive "*postal*" delivery service, combined with appropriate new DTN routing protocols would be a good solution. Indeed, the *Bundle Protocol*, [18], under development in the Internet Research Task Force (IRTF) Delay-Tolerant Networking Research Group (DTNRG) is extracted from the heart of the E-Mail principle to address the above-discussed issues and seems to be the most mature among other DTN protocols being under study at least until the time this manuscript was written. Thus, in turn, we will tend to give a tutorial-based version of this protocol's description and further discuss our views of its problems and appropriate solutions where those are possibly applicable.

B. Protocol basics:

The essence of the Bundle Protocol lies in the fact that all information data and control signals are combined in a single

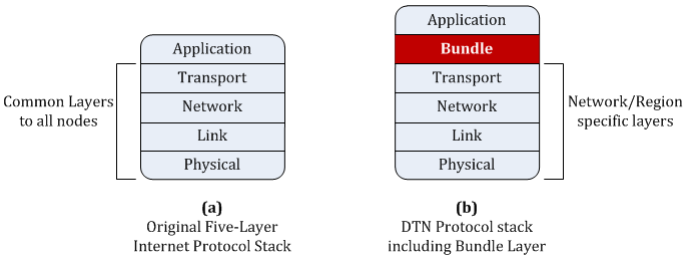

Fig. 11. Original and modified Protocol Stack.

atomic entity, called *bundle*, which is transmitted across a DTN.

1) A bundle's day out: Figure 10 illustrates a typical medical application where sensors held by different people/objects tend to monitor the viral activities in a small city. Figures 10-(a) through 10-(f) are self-explanatory. Some minor elaboration is required for parts (g) through (j) of the figure. In (g), it so happens that vehicle A, a taxi car driving on a certain lane, is passing by the construction site.

Beacons from a node mounted over A awakens all nearby sensors with each of which is being assigned a time slot. Sensors in turn transmit their bundles to A, particularly, our tagged bundle from the engineer's sensor. A accepts the custody of the bundles ($i.e.$ it takes the responsibility to store a copy of them until they are received and acknowledged by the DTN Gateway at the hospital's headquarter). As A continues to travel in (h), it meets on its way another vehicle, B, to which it transmits our important bundle possibly with others (assuming no occurrence of irregular events such as transmission failure, congestion, etc). It so happens that B is a taxi car on a different lane that is scheduled to arrive at a taxi stop right in front of the hospital in about thirty minutes and so our bundle arrives in (i) to a DTN gateway at the hospital, which in turn generates ACK bundle(s) indicating the admission of the custody of our bundle (and possibly others). ACK(s) will most probably reach A through different paths some hours later so that A may delete their copies freeing up storage space. At this point, the bundles didn't reach their ultimate destination yet, the doctor's PC. However, the DTN gateway notifies the doctor of the arrival of the bundle via an e-mail message that appears on the doctor's PC screen. Unfortunately, the doctor happens to be performing a surgery and so the bundle remains stored at the DTN gateway. A couple of hours later in (j), the doctor finishes his surgery and goes back to his office. Excited to read the data stored in the bundle, he opens his mail and thus acknowledges reception of the bundle. The bundle is finally received by its ultimate destination and the journey ends there within several hours after the bundle was first generated in (e).

2) The bundle layer: An overlay located in between the application layer and the transport layer. Figure 11 shows the modified five-layer Internet protocol stack that includes the bundle layer. For further illustration, assume that an application running on the PC of a scientist in the research lab on Earth (source, S in Figure 6) is sporadically communicating with its peer on a node in the Marsian Internet cloud (destination, D). Figure 12 illustrates this communication process from the perspective of inter protocol layer communication. Generally, all nodes may be mobile. Figure 12 only shows two gateway instances and only one intermediate node ($i.e.$ router or host) between them. However, in an overlay network, there is no limit to the number of intermediate nodes as well as the number of gateways that may exist between two ultimate end points. The importance is that intermittent connectivity is also supported regardless of the underlying network types (enumerated in section I) and the transportation of a bundle is presumably reliable between each and every one of the bundle layer instances [15], [18].

Figures 11 and 12 show that layers underlying the bundle layer are all network specific. It is not that these layers uselessly exist in the DTN protocol stack. Their existence indicates that the bundle protocol by itself as specified is not sufficient to carry information across the DTN but it relies on a variety of delivery protocols ($e.g.$ TCP, UDP, IP, Ethernet, Serial communication lines, handheld storage drive, etc). Given the variety and the multitude of protocols residing in each of them, those layers are referred to as convergence layers [13], [14], [15], [18] and are equipped with a set of protocol-specific Convergence Layer Adapters (CLAs) that are responsible of guiding DTN bundles through their corresponding protocol's different procedures, [2].

3) Names, Addresses and the Binding Process: The traditional Internet's operation revolved around the adoption of names instead of addresses to identify objects ($e.g.$ search engines, page caches, etc), [1]. Nevertheless, addresses are still used in the routing process as a reference to a given computational resource ($e.g.$ a particular server). Therefore, a mapping function, the Domain Name System (DNS), was introduced to translate names to addresses.

Similarly, in DTNs, nodes are identified by Endpoint IDentifiers (EIDs) (of maximum length of 1024 bytes) that can viewed as URIs, [13]. A single EID may either refer to a single node or a group of nodes where, in the latter case, it is intended to support multicasting. The DTN registered URI scheme is simply "dtn:" where the exclusion of any addressing is represented by a null EID as follows "dtn:none". Whatever comes after the URI scheme ($i.e.$ dtn:) is referred to as the Scheme-Specific Part (SSP). Several works are in progress for defining SSP rules to support anycast or nearcast identifiers (respectively [38] and [39]) or flat namespace probably employing 48-Bit MAC addresses [40]. Kevin Fall et. al. in [41] present an early thinking about this subject and specifies the syntax of new DTN URIs followed by a discussion of the resolution of EIDs. The matter is still subject to extensive debate and far from being finalized in the near future. From the already published material it's clear that EIDs have a relatively complex structure and people are pulling the rope to whether EIDs should be used as names or as addresses, [13].

Our interpretation of EIDs comes in favor of utilizing EIDs as names for two reasons. The first is obvious: as we have mentioned earlier, they are syntactically URIs. Second, we also mentioned in footnote (2) that URIs are names that only identify a particular resource rather than define its location

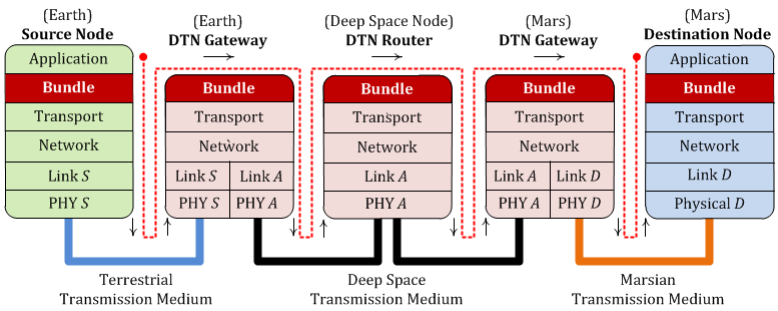

Fig. 12. DTN protocol stack communication diagram.

and how to obtain it which is, from the understanding of DTNs so far, very compliant to the situation of DTN nodes. Take a look again at Figure 5. For the sake of the example, let's assume for instance a scenario that applies to this Figure where DTN is used to interconnect independent regional Internets. Some mobile nodes in each region may, at any time, move to another region. DTN actually relies on those nodes to carry bundles from one region to another. Each time a node, say D, arrives to a region, it registers as being part of that region's Internet and obtains an address ($i.e.$ an IP address). The registration process is outside the scope of this manuscript (refer to [1] for more information). Accordingly, D's address is highly subject to change from one region to another. However, if D is given a certain name ($e.g.$ $dtn : eagle$), this name remains fixed irrespective of the region D resides in. Intermediate nodes participating in the transition of bundles destined to D need only identify D by its name ($i.e.$ $eagle$). Nevertheless, mapping the name eagle to an IP address ($e.g.$ 192.168.1.2) using DNS within a specific region may still be possible. However, the point is that node D may first belong to a network in one region, say A. Then it might move from region A to region B and from region B to region C at a later time and so forth. Therefore an active routing component is required that is smart enough to interpret the SSP part of an EID $-$ a process known as binding $-$ and figure out how to direct bundles destined to node D. The late binding principle described in [13], states that an EID's SSP should only be resolved and mapped to an address that is the closest possible to the destination. Finally, we note the possible existence of different nodes in different regions with the same IP address but the reader should at this stage clearly see that this should not be problematic.

4) The notion of a "contact": As opposed to the traditional Internet, connectivity in DTNs is intermittent. That is, in a DTN, nodes are not continuously online. Hence, they are not always contactable at any time instant. In addition, communication in DTNs is subject to numerous time-varying constraints and is characterized by such begin/end instants, endpoints and directions and most importantly is link capacity and delays, [14], [15]. In addition communication in DTNs is not always bidirectional ($e.g.$ deep-space and acoustic networks, [13]). Therefore, we can represent a DTN by an abstract graph $G(N, L)$ where N is the set of nodes and L is the set directed links represented by dotted arrow lines as shown in Figure 13. The existence of a link $e(e = 1, 2, 3, ...)$

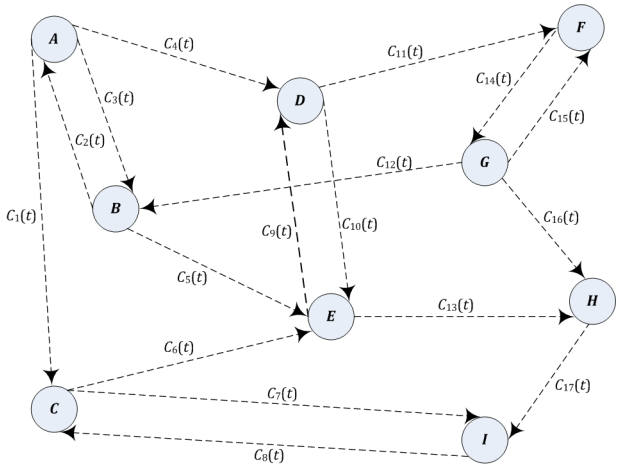

Fig. 13. DTN graph abstraction.

between two nodes indicates their ability to communicate as they opportunistically move in proximity of each other. In other words, when no communication is possible over link e, the capacity, $C_e(t)$, of this link is equal to zero. It is positive otherwise, and may fluctuate between a certain maximum and minimum when, for example, nodes are in relative motion. Thus, as defined in [13], [15], a contact is the time interval during which the timevarying capacity, $C_e(t)$, of a link "e" is positive, where time is measured starting from the instant data transmission on "e" begins. Furthermore, the volume of a contact is the total number of bits that can be transmitted from the source to the destination node during that contact.

Following the above definition, it is not the contact type by itself that changes but rather the connection type to which the contact corresponds. Therefore, we re-enumerate those types with a brief summary of each as described in [15]:

1) *Persistent connection*: A connection that is always available ($e.g.$ an Internet connection).
2) *OnDemand connection*: A connection that is initiated upon the request of a network entity to communicate with another entity but then becomes a persistent connection until it is terminated ($e.g.$ a dial-up connection).
3) *Scheduled connection*: A connection that is initiated at a particular time for a finite time duration ($i.e.$ contact) following a pre- established agreement between the communicating entities ($e.g.$ communication with an orbiting satellite).

4) *Opportunistic connection*: A connection that is established due to an unexpected opportunity ($e.g.$ Ad-Hoc connections).
5) *Predicted connection*: A connection that is established at a time instant, tpredicted, for a contact, $D_{predicted}$, computed based on some statistical information providing sufficiently elevated confi-dence that successful communication will occur at that instant and during that contact. Resources are provisioned for such a connection while always accounting for the likelihood it might not be successful or may not be established at all.

5) Timing and Synchronization: When you say a scheduled connection, you definitely implicitly assume that end nodes involved in establishing this connection have to be synchronized. This is especially true whenever either end is not always contactable, a typical situation in DTNs. Synchronization in this context can be quite a cakewalk: use synchronized clocks. As simple as it may appear, the bundle protocol assumes it is always the case. However, it is not as simple. In contrast, it is one of the most critical issues. Moreover, the bundle protocol simply takes it for granted. Therefore, our aim is to highlight what we believe are major problems in this corner and try to propose at least early thinking about some ways out.

In addition to the above, timing is in fact required for tracking down expired bundles. As we shall see later, a bundle creation timestamp field (holding an absolute time value) and the lifetime (holding a time period expressed in seconds) fields in the bundle header are conjunctively used for this purpose. The reader may have noticed the analogy to the IP's Time-To-Live (TTL) field that is used to avoid routing loops. Also, in case of possible fragmentation, the creation timestamp is the sole identification parameter used for fragment re-assembly analogously to IPv4's Identification parameter.

We believe, like others in [13], [42], [43] and [44], that imposing time synchronization as a requirement in DTNs is not as astute since DTNs are network scenarios prone to significantly long periods of nodal isolation/disconnection. The IPN is one example [12], [13], [43] and [45]. Even some terrestrial experimentations, [42], showed that the lack of tight time synchronization caused system clocks to significantly drift in such a way that bundle creation timestamps were completely out of synchronization resulting in significant bundle rejections due to early bundle expiries. To top the cake with some cream, imagine what would happen if nodes in such circumstances used bundles that they send using the Bundle Protocol to learn the time: Absolutely inappropriate and resource wasteful. The reason for this is the following: all such time-request bundles will be judged as expired and thus discarded. The authors of [13] and [41] expect some solutions to be based on the development of some kind of DTN time synchronization protocols. In [46], a sort of Interplanetary Timekeeping protocol based on the famous Network Time Protocol (NTP) is proposed.

6) Custodians, Custody Transfers and Reliability Issues: Traditionally, TCP provides reliable data transfer services to upper layer applications where only end nodes are responsible for acknowledging the reception of error free packets or requesting retransmission of those corrupted or lost packets. This is fine since Internet nodes are reachable most of the time. In DTNs, however, this is inefficient or simply impossible. For example, in a DTN sensor network, the source is typically a small sensor with limited storage capacity that is unable to store packets for long periods of time to handle possible retransmissions. Hence, a lost or corrupted bundle is highly likely to be unrecoverable.

For the above reasons, the Bundle Protocol presents a mechanism, called *custody transfer*, to convey the retransmission responsibility of a bundle (or possibly one of its fragments) that has not yet reached its ultimate destination, to a node other than its source. The node that currently holds the custody of that bundle is called the bundle's custodian. Ultimately, the custodian is the bundle's source. As time goes by, the custody may be transferred to other intermediate nodes. However, the next bundle's custodian candidate must meet the following requirements:

1) Be closer to the bundle's ultimate destination.
2) Certify long period bundle storage ability.
3) Certify the ability and willingness to strive for the ultimate goal: depositing the bundle at its ultimate destination.
4) Possess enough power to remain usefully active over long periods.
5) Be cooperative and take advantage of every chance to realize the ultimate goal.

At first glance, this custody transfer mechanism sounds appealing to solve DTN reliability issues. However, several problems are hidden behind this appeal. It is a new born concept, [2], [13], [14], [42]. Some believe that it provides less reliability than TCP [13], [42] and others see it as an optimization to the end-to-end reliability [14]. Let us look closely at how this mechanism works as we illustrate in Figure 14. This illustration is based on the description in [15] and [47]. Parts (a) and (b) of Figure 14 are straightforward. In (c) the source, S, encounters a node $I1$. S first transmits the bundle to $I1$ that, in turn, receives and stores it in its persistent storage space as shown in (d). In (e) considering $I1$ as a valid candidate custodian given the requirements listed earlier, S transmits a special request bundle to $I1$ asking it to take the custody of the bundle and starts a time-out timer. Note that if no reply is received from $I1$ before the timer expires, S then retransmits the bundle again followed by another request. Luckily enough, $I1$ accepts the custody of the bundle and returns an acknowledgement bundle back to S. Upon the reception of the ACK, S deletes the bundle from its buffer and completes the custody transfer process successfully. This process is continuously repeated until the bundle is received by the ultimate destination. One may argue as in [14] that only end applications know what they required. This is why actually an additional option enables the source to require an ultimate acknowledgement from the destination indicating the reception of the bundle. This is quite consistent with the end-to-end reliability approach.

After this description, several weaknesses in the custody transfer approach can be highlighted. It is mentioned in [14] that the custody transfer mechanism ensures reliability on a hop-by-hop basis rather than end-to-end. Note however, that,

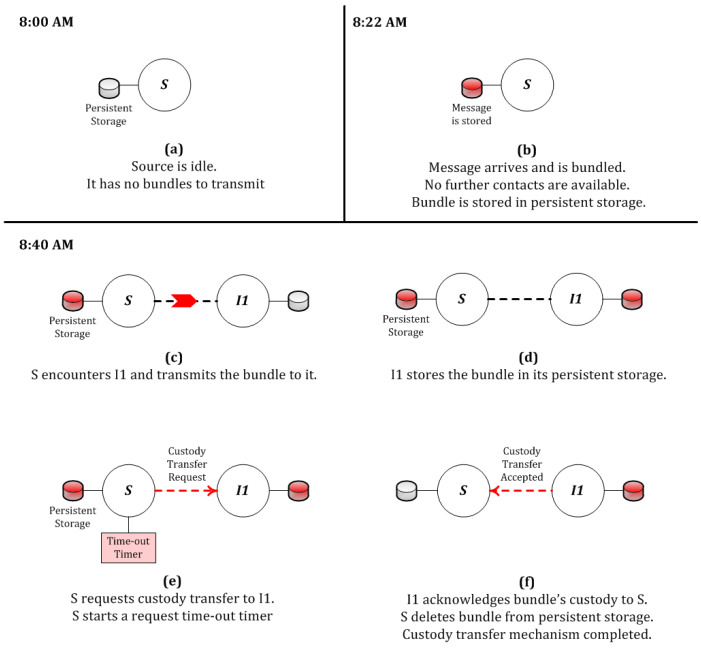

Fig. 14. Custody transfer mechanism.

throughout the entire process, we did not fall on any error detection process and the rejection of corrupted data. The first major weakness is that the Bundle Protocol, as described in [18], cannot determine if a received bundle is corrupted or not. The rationale behind omitting error detection is as presented by [15], [18] and quoted from [42]:

"*... not all applications require error-detection or data integrity applications can provide these features themselves*."

Another major issue is illustrated in parts (e) and (f) of Figure 14. Without chattiness the bundle's custody transfer would have never been successful. Had the one-way communication constraint been unavoidable, as is the case in the vast majority of DTN scenarios, the custody transfer mechanism would have failed as the source would have had no way to ever know that $I1$ has accepted the custody of the bundle. Ultimately, if the source is only a transmitter, then there is no way ever to inform it of the need for retransmissions.

We believe that a third major problem stems from the fact that custody transfer is a loose cooperative mechanism that relies primarily on the sole unjustified willingness of a node to accept or reject bundles' custodies being dedicated to it. In the presence of selfish and/or malicious nodes this increases the likelihood of bundle drops due to congestion, as congested custodians will not be able to convey bundle custodies forward and free up storage space for other newly incoming ones.

When carefully re-examining what happens in Figure 14-(e), S asks $I1$ to take the custody and starts a timer whose expiry causes S to retransmit the whole bundle again and re-ask $I1$ for custody transfer. But what if $I1$ is not willing to take charge of the bundle? It is not specified and unclear in both [15] and [18] how this case is being taken care of. If no action is taken by $I1$ ($e.g.$ reply with rejection) then S will be retransmitting the bundle again and again endlessly until $I1$ is no more in contact range. This is a serious problem as valuable bandwidth is inefficiently lost during the repeated retransmission of that same bundle.

Finally, as mentioned in [13], a candidate custodian, X, may satisfy all the above-listed requirements except (1.). If no candidate satisfying (1.) can be found, then it might be sometimes advantageous to send bundles to X so that it can handle possible retransmissions. The question as to how to route bundles this way remains without an answer.

7) Reporting: We observed earlier that for a successful completion of a bundle's custody-transfer, a negotiation process involving round-trips is required. In DTNs this is not allowed. Researchers have proposed what we also believe is not a solution but rather an early thinking of a solution idea: let there be an EID field ($e.g.$ report-to) in the bundle's header indicating the nodes to which special bundles called reports should be sent when required, [15], [18].

The DTN architecture and Bundle Protocol specifications differentiate between six types of such reports as listed in Table I. To request a report, its corresponding flag in the bundle's header is set. To avoid the possibility of infinite report clouds resulting in crushing Denial of Service (DoS) attacks, the specifications explicitly specify that no reports be generated as results of other reports.

Under some routing schemes such as Epidemic Routing (ER), as explained later in section IV, if we assume a bundle's forward flag is set, then upon the forwarding of each and every copy of the bundle (or possibly copies of its fragments) a report will be generated and sent to the bundle's custodian creating a storm of unicast report traffic to that custodian. This is yet a potential problem of the reporting scheme.

8) Congestion and Flow Control: While the authors of [14] defined *Flow control* as the limitation of the sending rate of a DTN forwarder and *Congestion control* as the handling of contention for persistent storage at a DTN node, DTN Congestion and Flow control is quite an immature area. This has been openly admitted in [15]. The DTN Research Group by itself still did not reach an agreement on what approach to follow. It is worthwhile taking a general and brief idea so as to explain what is going on. The definitions given in [15] are general and may be applicable to any network type. The authors of [2], [13] and [14] refine these definitions and propose DTN-oriented congestion and flow control approaches.

As far as flow control is concerned, the DTN architecture relies mostly and attempts to benefit from the mechanisms implemented in underlying region-specific transport protocols such as TCP, Ready/Clear To Send (RTS/CTS), different admission and rate control schemes, X.25, XON/XOFF, etc.). In light of the related section in [15], a DTN node with a message to send clearly assumes the presence of flow control mechanisms to ensure reliable message delivery.

Congestion control, however, is a different story and is much more complicated due to the following reasons:

1) Unavailability of near future contacts resulting in bundle accumulations in nodes' buffers.
2) Bundles with accepted custodies remain permanently stored until either their expiration or the occurrence of

TABLE I
BUNDLE REPORT TYPES.

Report Flag	Report Description
Custody Accepted	Sent by a candidate custodian upon accepting the requested custody transfer of a bundle.
Bundle Received	Sent upon bundle arrival to destination node.
Bundle Forwarded	Sent upon forwarding a bundle.
Bundle Delivered	Sent upon a bundle's delivery by the Bundle Protocol to the application at the destination.
Bundle Deleted	Sent when an expired bundle is deleted.

an unusual event ($e.g.$ node destruction) resulting in their deletion.

The author in [14], explains that a priority-based queuing system is adopted for storage allocation. Multiple bundle priority classes are considered. However, incoming bundles are first classified within two categories, namely: a) Custodially Admitted Bundles (CABs) and b) Non-custodially Admitted Bundles (NABs). CABs are buffered ahead of NABs. Furthermore, bundles of each category are sorted depending on their priority classes. Moreover, the author specifies that large bundles are never custodially admitted. Finally, all CABs are drained first followed by NABs. However, it is important to note that the highest priority bundle within each category is cleared out first followed by other lower priority ones. The number one problem in this approach is that there is no specification of when a bundle, in terms of size, is interpreted as large. Problem number two is that lower-priority bundles may arrive to a node and be custodially admitted earlier than higher-priority ones. They will therefore occupy buffer space in such a way that future incoming high-priority bundles may not be admitted due to storage space unavailability. Problem number three occurs whenever the contact during which CABs are to be drained still did not start while NABs can be immediately cleared out during a present opportunity. Given that CABs have must be cleared first, NABs are simply stuck behind them.

To leverage congestion, any one or combination of the following solutions proposed in [2] may be used:

1) Discard unexpired bundles (questions the network's predictability from the end-user point of view).
2) Bundle displacement (quite a good approach if storage exists at nearby suitable[1] nodes).
3) Accept no more custody for incoming bundles (causes custody transfer backlogs and hence probably congestion at upstream nodes).
4) Accept no bundles at all (may also cause congestion at upstream nodes).
5) Class-of-Service based buffer space reservation (increases lower-priority bundle drops as well as upstream congestion).

Note that discarding custodially accepted bundles conflicts with the aimed DTN trusted delivery abstraction and is therefore omitted from the above list. We have also omitted the option of discarding expired bundles because those latter are going to be discarded anyway and may not be there by the time congestion occurs. Finally, we note that all the above points result in performance degradation for their obvious reasons.

[1]Nearby nodes where buffer space is available may not be suitable custodian candidates for custodially accepted bundles being moved

9) Bundle Fragmentation: In DTNs, bundles may traverse various region-specific networks that are typically heterogeneous. One of the implications of hetergoenity is the limitation on the maximum size of their respective Protocol Data Units (PDUs). Since bundles are to traverse such networks, then surely they are subject to such restrictions. This is a primary motivation to allow for bundle fragmentation. Another motivation is the fact that in some DTN scenarios ($e.g.$ Galileo probe to Jupiter) channel capacity is so low that the bit rate does not exceed couple of hundred bits per second. In such scenarios, the transmission of complete large bundles will never be successful especially when contacts are relatively short as compared to propagation delays.

In some DTN scenarios, connection disruption rates are relatively low. As such, the underlying reliable network protocols ($e.g.$ TCP) are able to support relatively larger bundles (as compared to those of the Galileo example) and recover from packet losses and corruption. In such cases fragmentation will not occur at the bundle layer. In some other cases, connections experience intensive disruptions that even the underlying reliability mechanisms fail. Let us not forget that sometimes the underlying protocols are also not necessarily reliable ($e.g.$ UDP). It is in such cases where we might frequently face a situation where some bits and pieces of a bundle may be successfully transmitted but not the complete bundle. The net result is complete bundle transmission failure. Of course the retransmission attempt of the entire bundle squanders precious resources and may by itself fail again. This is yet a third motivation for bundle fragmentation.

Two types of fragmentation are defined for DTNs, [2], [13], and [15]:

1) *Proactive Fragmentation*: The subdivision of a large bundle into smaller size fragments prior to the establishment of a connection of known duration and channel capacity ($e.g.$ Galileo probe to Jupiter). This type of fragmentation can also be used for adapting bundles to lower-layer message oriented transports, [2].
2) *Reactive Fragmentation*: The subdivision of a large bundle into smaller size fragments when lower-layer protocols indicate that the large bundle was only partially successfully transmitted.

It is worthwhile noting that reactive fragmentation is dynamic and is therefore more challenging to handle as it conflicts with other custody transfer mechanisms and other bundle security issues such as authentication, [13].

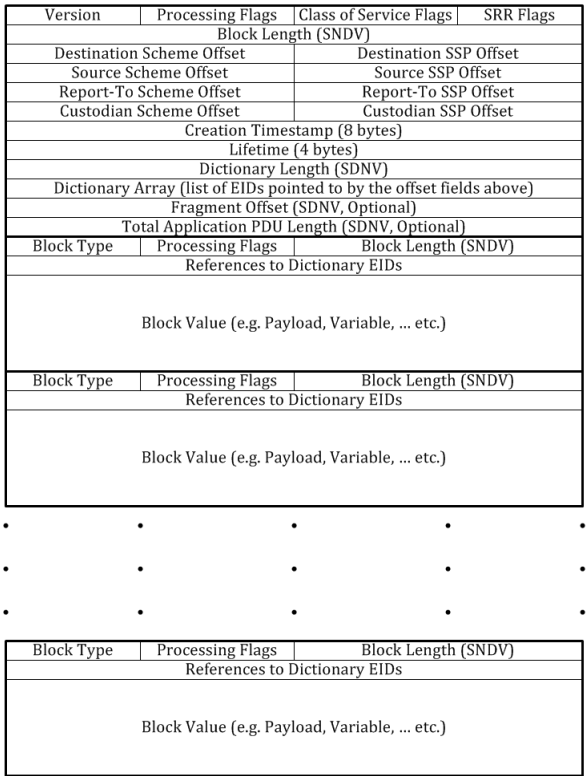

Fig. 15. Custody transfer mechanism.

The support of the bundle protocol's fragmentation is achieved in the same way as in IPv4, [2]. A *Fragment Offset* header field indicates the fragment's offset and length relative to its position in the original bundle. Bundle fragments are only re-assembled at the ultimate destination, [15]. A common identifier including the sender and receiver's EIDs and the creation timestamp is used as the sole identification of the belonging of fragments to the same original large bundle. No specification pertaining to the control of fragment routes takes place in the Bundle Protocol. Intermediate fragment re-assembly (other than at the final destination) is therefore possible. The custody transfer mechanism is enhanced to support fragmentation. However, for the same bundle, some fragments may handled by one custodian while others dealt with by another custodian. Hence, there may be multiple custodians for the different fragments of the same bundle, [2], [13].

Some security requirements may sometimes require a bundle to be digitally signed where this signature is to be verified for cryptographic correctness by each intermediate node the bundle arrives to before reaching its final destination. Fragmenting the bundle in this case will render this type of verification impossible. For this purpose, the Bundle Protocol incorporates a *do-not-fragment* header field that prevents any fragmentation of the bundle.

10) Basic Bundle Structure: Writing about the structure of a bundle is quite descriptive and may follow exactly what has been repetitively written in [2], [13], [14], [15] and [18]. For the sake of completion, we will only show in Figure 15 the basic bundle structure (headers and payloads) leaving out the details about why these were represented that way. For further information in this regard, the reader is invited to go over in the above references.

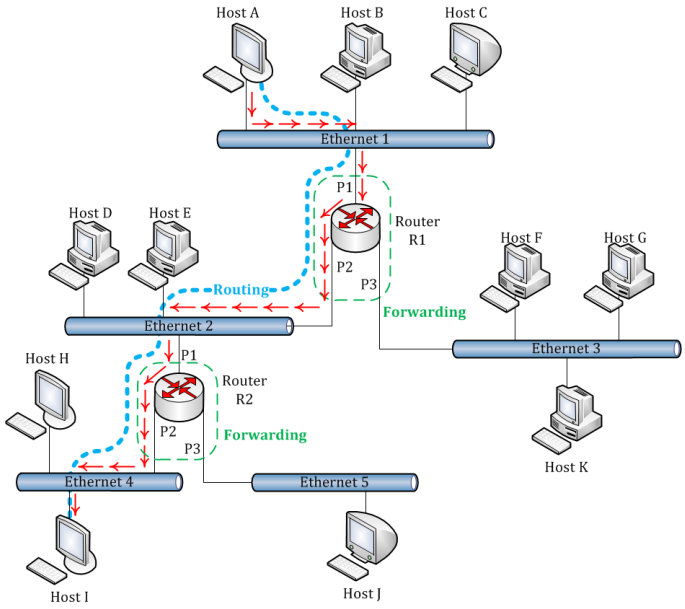

Fig. 16. Routing and Forwarding in typical Internet networks.

Many of the bundle block fields are represented using the Self-Delimiting Numeric Values (SDNV) flexible encoding technique for efficiency purposes [13], [18]. Most of the fields seen in Figure 15 are self-explanatory. The one attractive aspect in this structure is the dictionary. Notice that not only source and destination EIDs are included but also EIDs for custodians, and report-to nodes. However, it is often the case that custodians be also report-to nodes since they are the sole nodes responsible of the bundles and are the only ones able to react in case something wrong happens. So instead of including the EIDs of such nodes twice as custodians and report-to nodes, they will only be included once in the dictionary and pointed to by the custodian and report-to fields.

V. Routing in Delay-Tolerant Networks

A. Preliminaries

The so-called "*Routing*" in DTNs has been a very widely addressed matter that almost distinguished itself as an independent research area where a vast and rapidly increasing amount of works continue to appear. Nevertheless, in light of our rigorous study of DTNs, we would like to share our own understanding of routing in this type of networks. Note that the difference between the *forwarding* and *routing* functions of the traditional Internet's network layer has been extensively clarified in [1].

Figure 16 shows a simple network with two routers, $R1$ and $R2$, and several hosts. Suppose that host A is communicating with host I. The figure shows how the packets transmitted by A pass through routers $R1$ and $R2$. In fact, when A sends packets to I, router $R1$ captures them on port $P1$ and then uses its routing tables to forward them to port $P2$. The same occurs with router $R2$ as it forwards the packets to Ethernet network 4 where they are finally delivered to I. The orchestrator of this

entire process is the routing protocol that makes it possible to find a least cost path from the source router to the destination router. The details of how routing tables are constructed and used and how the entire routing process is conducted are outside the scope of this manuscript. Nevertheless, we will visit some of the basic tasks of routing protocols and elaborate on their achievability under the extreme limitations of DTNs. In brief a good routing protocol must be able to:

1) Rapidly and reliably deliver packets.
2) Adapt to network topology changes resulting from nodal or link failures.
3) Adapt to the variability of traffic loads.
4) Temporarily route packets away from congested links.
5) Determine the network's connectivity.
6) Avoid routing loops.
7) Minimize routing overhead.

Let us wear the shoes of a DTN source node, S, that has some information destined to a certain DTN destination node, D. S will generate the bundle. However the only information that S knows is the identity of D. It is often the case in DTNs that at the moment the bundle is generated at S, this latter happens to be disconnected. As such, there is no possible way the bundle can be sent further and must definitely remain stored until an opportunity to send it out occurs. Yet if being so optimistic we assume that S was able to send out the bundle right away to a next hop. Then we may ask:

1) Will there also be an immediate opportunity at this next hop so that the bundle doesn't get stuck there?
2) Is this currently available opportunity the best?
3) Are there going to be even better opportunities in the near future?
4) If the bundle is sent out, is it going to reach the intended destination?

The answer to those questions, as ignorant as it may sound: *There is simply no way to know*! But keeping the bundle at the source will not be a better choice either. If at each occurring opportunity the source decides not to send the bundle out so that maybe better opportunities come in the future, then most probably the bundle is going to expire at the source and thus ends up being discarded. Therefore, sending the bundle forward increases, even if slightly, the likelihood of its delivery. The same situation is highly possible to occur at the bundle's second hop and further next hops until it is ultimately delivered. Some historybased approaches [49], [50], [51], try to tackle this issue by accounting for the so-called delivery probability. If two or more next hops are simultaneously available, then forward the bundle to the one that has the higher delivery probability. The common between all those approaches is that a next hop choice is made only based on the probability that this candidate 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. This is not to add that such history based schemes make use of a learning process that takes a considerable amount of time in the presence of excessive delays. Add to all this that nodes are 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. A recent work in [52] examines the effect of topology knowledge on opportunistic forwarding and show that partially propagating topological information over a number of k hops is beneficial in performing overhead versus delivery latency optimization. However, as the authors said, such a protocol might not be optimal but the ease of its deployment and its benefits are appealing. Their results clearly show that although latencies significantly decrease as k increases, overhead will drastically increase. This is something that is strictly undesirable in DTNs. On a different note, in DTNs, only those nodes accepting custody of a bundle are those that are promising to do their best to deliver that bundle. In fact, a bundle is not guaranteed to be delivered since it might expire and be discarded if the situation is so extreme that no opportunities occur before expiry. The situation can be made worse when not all nodes are willing to accept the custody of a bundle. This is especially true since custody acceptance is only left to the consent of the node and does not appear as part of an administrative decision imposed by well determined protocol rules. To date, all the so-called "*DTN routing protocols*" offer some metric evaluation on a per-hop basis helping a node choose the appropriate next hop that is only currently best to forward the bundle to. That is, the single hop forwarding decision is based on a given metric evaluation, which is only valid at a current epoch. This discussion finally leads us to question the suitability of adapting the term "routing" to DTNs.

To accurately answer this question, we will simply revisit the seven main tasks of a routing protocol but now in the context of DTNs:

1) Packet delivery in DTNs can in no way be as fast as it is in typical Internet. It is possible sometimes to perform next hop selection while aiming at minimizing the delay but still delay will remain comparatively much larger than in traditional Internet. Delays are therefore present and it is up for the applications to tolerate them and the protocols to adapt to them. However, the main concern remains reliable delivery.
2) As links fail, nodes become disconnected and vice-versa. This will incur topology changes and ultimately partitioning, in which case a packet's next hop becomes unreachable with no other alternative. In DTNs, this situation is the norm rather than the exception. As a result, instead of dropping a bundle, nodes will store it in their buffers until any forwarding opportunity presents itself. The node is then said to store, carry and then forward the bundle, a functionality that is supported by the bundle protocol.
3) In DTNs, adaptation to traffic loads is not achievable as there are few available choices to do so. As mentioned earlier, topological knowledge is sometimes totally unavailable. Nodes are left out to act as per their own incentives and routing decisions are based on some metric evaluation imposed by the DTN routing protocol. At this point, one can argue that network load can be taken as a routing metric. But again, we must not forget that network state information is unavailable in

frequently disrupted networks so it doesn't make sense for network load adaptability to be a requirement in DTNs.

4) As a continuation of (3.) one could argue that in some scenarios nodes would only send bundles to next hops that have the highest delivery probabilities. As a result, only part of the network's nodes will be loaded with bundles while others will not. This can be seen as a network load-balancing problem but we can arguably refer to it as a congestion problem, since those nodes always receiving bundles will easily suffer from buffer exhaustion. In this context, buffer space can be efficiently managed by having a forwarder refrain from forwarding a bundle to a congested next hop. However, notice that there is a huge difference between a congested link where nodes contend to transmit bundles and a congested buffer where bundles contend for the use of storage space. In DTNs, as in any wireless network, the channel is a wireless broadcast channel. If more than one node are in the range of each other (a very rare scenario in frequently disrupted networks) transmissions cannot be diverted to other links as, after all, only one link is present. In such a situation, when a node transmits it will be heard by all its neighbors. The traditional hidden/exposed terminal problems may occur and solving such problems may be done in the usual ways used in traditional Internets.
5) In DTNs, there are scenarios where connectivity can at most be predictable up to a certain point. Those scenarios are referred to in the open literature as deterministic DTNs, [53], [54], [55]. An example would be the communication with a low Earth orbiting satellite. But even in such cases, a satellite may drift a little off its orbital trajectory and thus may not be present at the predicted instant resulting in connection failure. In the case of stochastic DTNs, [49]– [51], [56]– [76], determination of connectivity and even predictability is quite challenging.
6) In the traditional Internet, Link-State routing protocols such as Open Shortest Path First (OSPF), Intermediate System to Intermediate System (IS-IS) and Optimized Link State Routing (OLSR) avoid routing loops by flooding topology changes to all nodes in a routing area, [1]. This is something practically impossible in DTNs. On the other hand Distance Vector protocols such as Border Gateway Protocol (BGP) and Enhanced Interior Routing Gateway Protocol (EIRGP) have built-in routing loop detection mechanisms. For example, BGP lists the autonomous systems that a route passes through, [1]. However, since no end-to-end route for a bundle is known in DTNs, such mechanisms cannot be applied. In addition, the bundle protocol does not have any capability to avoid routing loops and is not extendable to list the nodes a bundle passed through. Even if that was possible, on the long run this it becomes inefficient if the number of nodes from source to destination increases with time. In summary, no mechanism has been defined in DTNs to avoid loops.
7) One of the main concerns in DTNs is avoiding inter-activeness. This is mainly due to the fact that chatty protocols are the major source of overhead. In addition, information exchange that takes place to help nodes make better decisions about next hops, incurs additional routing overhead. This is why sometimes one-way communication is enforced. However this reduces reliability as we have discussed earlier.

It is clear from our discussion that all of the tasks related to routing are not achievable in DTNs. Therefore we cannot speak of a routing protocol but rather an opportunistic forwarding algorithm based on a set of next hop selection rules aiming at delivering a bundle to its intended destination.

B. Related Surveys

Z. Zhang, in [10], exposed a wide survey of the unicast routing schemes that have been published up until May 2006 including:

1) Deterministic schemes where the future network's state/topology is predictable, hence allowing message forwarding to be scheduled ahead of time. The space-time, tree-based, and modified shortest path are examples of deterministic routing schemes.
2) Stochastic schemes where the future network's state/topology is completely unknown and hence no pre-scheduling of transmissions can be done. The following routing schemes fall under this category: Epidemic and randomized flooding, history-based, model-based, controlled- movement-based, and coding-based schemes.

In addition, another survey by Z. Zhang co-authored with Q. Zhang, in [11], addresses further developments in this area up until May 2007 and pertaining to three different sub-categories:

1) *Message Ferrying Approach*: As network partitioning occurs, additional special mobile nodes called Message Ferries (MFs) are injected into the network. MF movements span the entire network area with each one of them mainly responsible for carrying bundles from nodes in one partition to nodes in another partition. The challenge is to determine the number of required MFs and determine the route of each so as to reach a certain optimal objective.
2) *Inter-Region Routing*: In regional networks such as the one illustrated in Figure 6, routing bundles from one region to another encompasses several challenging problems such as naming, binding, route selection, protocol translation and reliability control.
3) Multicast Routing: Dissemination of bundles to node groups. In the DTN context, a fair example would be disaster recovery scenes where it is particularly essential to distribute critical information about casualties and possible hazardous events to rescue teams. DTN Multicast semantics definitions as well as the determination of the suitable forwarding instants persist as challenging problems in this category.

Being identified as a key element of the DTN architecture, the so-called routing in DTNs is still attracting the attention

of a huge community of researchers. In the following subsection, we aim at surveying works done in the context of DTN routing between mid 2007 and June 2010, the date this manuscript was written.

C. Opportunistic Forwarding

In a DTN, assume that at a certain time instant t_0, a node N happens to be disconnected. A bundle B, held by N, and ultimately destined to a set, D, of one or several nodes, obviously has an undetermined next hop at t_0. At future points in time, say t_1, t_2, etc, different sets of one or several nodes may respectively appear as N's one-hop neighbors. Therefore, given a set of constraints ($i.e.$ immutable environmental limitations, latencies, channel conditions, nodal power availability, etc.), the DTN forwarding problem consists of determining: (i) B's best next-hop among the currently available or upcoming neighbors of N and (ii) the most suitable time instant at which B must be forwarded to that next hop, in order to maximize: $a)$ its likelihood of successful delivery to D and $b)$ its contribution in the overall network's performance optimization of a pre-defined metric.

In [52], the authors investigate a class of opportunistic forwarding algorithms in duty-cycling, energy constrained, wireless ad-hoc networks[2]. As a result of duty cycling, the network's topology becomes highly time varying. It is argued against the dissemination of network state information especially in low duty cycle[3] networks as it may become obsolete by the time it is received by certain nodes. In their previous work, [78], the authors analyzed opportunistic forwarding ($i.e.$ forward to the first available opportunity) and showed that it suffers high end-to-end latencies due to random walk. In this present work, they first derive exact expressions for such mean delays in a Manhattan routing scenario as a function of grid size n and duty cycling probability p. Next, they observe that link state information dissemination to a limited scope of nodes minimizes control overhead and avoids congestion resulting from broadcast storms. Based on this observation, they propose to partially disseminate network topology information ($i.e.$ limited to k hops) as an effective solution to reduce end-to-end delays. However, nodes that are k-hops farther away from an arbitrary destination still lack information about how this destination can be reached. This is why, the authors came out with a hybrid protocol, RANDWLS, where opportunistic forwarding is used until a node that is at most k-hops away from the intended destination is reached. From that point on, shortest-path routing takes over.

The overall of this work sounds appealing and non-complex especially when it comes to the implementation of RANDWLS which is quite easy even if it turns out not to be an optimal protocol. However, we strongly disagree on the validity of both the proposal and the protocol in DTNs extreme environments due to the following reasons:

1) In a highly disruptive network, some link state information, destined to a node N_k that is k hops away from a node D, is highly likely to become obsolete due to a disruption that occurs while in transit. The reason is that the immediate next hop at the point where the disruption occurred, after recovery, might not be the same as the one that was before the disruption.
2) In DTNs where signal propagation latencies are significant ($e.g.$ IPNs), it is impossible to disseminate topology information as this latter might become obsolete even before it reaches the immediate next hop.
3) In low duty cycle networks, a sender may have to wait long before its intended receiver wakes up (assuming that such knowledge is available or at least predictable). It may happen that the topology information destined to this receiver becomes obsolete by the time this latter wakes up.
4) In addition to the point in (3.), usually in predictable scenarios, bundles are given large lifetimes in such a way that they do not expire before they are able to propagate further. This creates an efficiency problem if the information in such bundles becomes obsolete but the bundle itself did not expire meaning that it will be inefficiently transmitted when the opportunity comes.
5) It is inappropriate to use existing Internet's shortest-path algorithms in DTNs as for the so many reasons discussed earlier, they are fated to failure. They might only be used in the specific scenario of regional networks that we discussed earlier and illustrated in Figure 5 where only the network within a particular region's boundaries is highly connected.

V. Conan et al. in [77], propose a single-copy multi-hop opportunistic routing scheme for sparse DTNs based on a fixed point recursive process: The MH^*, an optimal multihop relay scheme that aims at maintaining a low network load. It uses as only inputs the estimates of the average nodal inter-contact times. On one hand, the authors observed that the mean inter-contact time varies for each and every arbitrary node pair in the network. This is how, by modeling the pairwise contacts as independent Poisson processes with different parameters, they aim at capturing those variations. As a consequence, inter-contact times are exponentially distributed with different means. While the memoryless property of the exponential distribution offers a lot in terms of tractability [79], [80], such a model is rudimentary as it might not match all realistic scenarios specially those incorporating a mixture of light-/heavy-tail distributions. On the other hand, the authors capitalize in this work on the advantages of transitivity[4], an observation that was first introduced by Lindgren et al. in [50]. To build their MH^* scheme based on the above observations, the authors start by first considering the two-hop relay scheme of Grossglauser in [57] that we briefly summarize as follows:

Consider a total of n nodes in the network. One is a source S that carries a bundle B (marked in red in part (a) of Figure 17), one is a destination D and the remaining $I_k (1 \leq k \leq n-2)$ are intermediate relay nodes. In a first phase, S waits for any arbitrary forwarding opportunity to occur. This latter

[2]In such networks, node transceivers are independently shutdown during idle times to preserve valuable battery power and then again powered on asynchronously as a communication opportunity becomes available.

[3]Nodes stay off most of the time.

[4]For example, if a node X frequently encounters node Y which in turn frequently encounters node Z, then this latter can be seen as a good candidate to forward bundles to X.

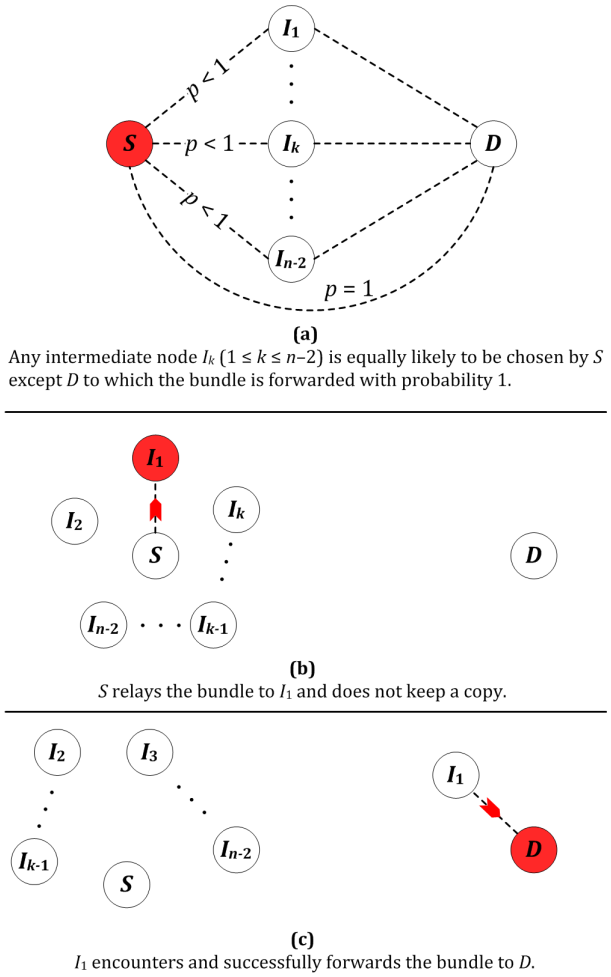

Fig. 17. The original two-hop relay strategy.

may consist of a single or multiple nodes. When D is among those nodes, S only forwards to D with probability $p = 1$. If the opportunity consists of a single node, I, then S will also forward to I with $p = 1$. Otherwise, any intermediate node I_k available in the occurring opportunity is equally likely to be selected as a relay with probability $p < 1$. Assume a node I_1 has been selected as the next relay node. Then, in a second phase, S relays B to I_1 without keeping a copy of it (part (b) of Figure 17). I_1 will keep on holding B until it meets with D to which it forwards the bundle and completes a successful delivery (part (c) of Figure 17).

The authors refer to this strategy as 2-MH ($i.e.$ the two-hop version of a potentially multihop forwarding scheme). Second, to optimize the above 2-MH scheme, S only relays bundles to a subset of its neighbors with a view to minimizing the average delivery time. The authors refer to this optimized version as 2-MH^*. Finally, the recursive application of 2-MH^* results in a fixed point, which minimizes the delivery latencies when more than one intermediate relay steps are considered. It is justifiably argued that the number of steps is finite and thus the recursive process is proved to have a polynomial running time. Also, since bundles are forwarded only to nodes that are closer to the destination in terms of expected delivery time, then the scheme is shown to be loop free.

Extensive simulations were performed to evaluate the performance of the scheme using three basic mobility traces, namely: Dartmouth[5], iMote[6] and MIT[7]. The iMote and MIT traces sound suitable to model DTNs due to the short range of Bluetooth (about 10 to 12 meters) that can emulate in one way or another the disruptions between distant devices. However, these traces underestimate the number of contacts due to the periodicity of Bluetooth scans. The use of Dartmouth traces to perform simulations is, in our opinion, inadequate as the Dartmouth College network is not a DTN. Indeed, the access points deployed all over the campus ensure that contemporaneous end-to-end paths always exist between any source/destination pair. It is clear though that the authors invested lots of efforts to adapt the Dartmouth traces to DTNs through filtering and judiciously assuming that two nodes are in contact only if they are simultaneously attached to the same access point. Despite these efforts, we believe that such adaptations are unrealistic and hence are not accurate especially that laptop mobility does not necessarily emulate human mobility. These non-realistic traces may, nonetheless, be justified by: a) the inexistence of real DTN traces to perform simulations on, b) the elevated cost of deploying a real DTN testbed and c) the non-existence of a suitable simulator accounting for all DTN characteristics.

Although named as "Congestion Aware Routing", the work in [81] is an example of a next hop selection forwarding algorithm that uses Multi Attribute Decision Making (MADM) concepts for next-hop selection over congested deep space networks. Generally speaking, congestion in DTNs has received very little attention especially when formulated as a joint optimization problem involving several other performance metrics such as message completion rate, data transfer time, power consumption, buffer occupancy, buffer filling rate and available bandwidth. The authors of [81] target such formulations and propose some possible resolution criteria. In reference to [82] and [83], the authors assume that the Bundle Protocol (BP) is directly implemented on top of the Licklider Transmission Protocol[8] (LTP). Herein, they do not make use of BP's optional custody transfer mechanism and therefore communication reliability is assumed to be only ensured by the underlying layers.

The authors adopt a reference network scenario incorporating three different types of nodes namely: a) regular planetary sources and destinations deployed in distinct planetary regions, b) backbone interplanetary nodes that act as relays forming a wireless mesh topology connecting the planetary regions all together and c) gateway nodes through which planetary nodes connect to the backbone nodes. Figure 18 is an imaginary representation of this scenario used for illustration purposes only. In the above-presented scenario, reliable data transfers are taken care of by LTP. So, upon the detection of LTP data unit losses, a recovery procedure, consisting of a Selective-Automatic Retransmission reQest (S-ARQ), triggers the retransmission of the lost units. In the context of that

[5]Traces collected from the Wi-Fi network of Dartmouth College.
[6]Bluetooth contact loggers at the INFOCOM workshop.
[7]Bluetooth contact loggers at the MIT Reality Mining experiment.
[8]LTP [82], [83], is a point-to-point protocol responsible for reliable data transmission over deep space links.

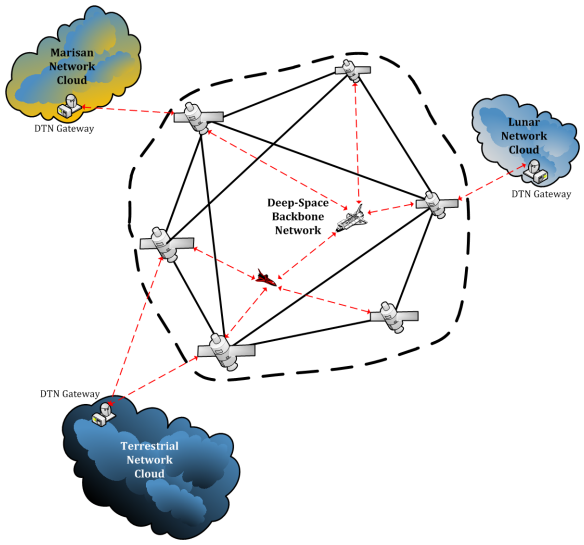

Fig. 18. Real-life realization of the reference scenario of [81]. Planetary sources are implicitly assumed to exist internal to each network cloud.

same scenario, the bundle protocol would be responsible for performing data store-carry-and-forward operations among the different regions.

In light of the above scenario the authors highlight the topicality of the bundle layer buffer space and declare it as a critical resource. As a matter of fact, the above scenario is all about communication in deep space where propagation latencies are significant. LTP, in turn, breaks down large bundles into smaller blocks to fit the Maximum Transmission Unit (MTU) of a link. Applying SARQ means that the originating node would have to keep a bundle stored up until the receiving node acknowledges reception of all pertaining blocks. Meanwhile a timer is started for each transmitted block. Upon receiving the last block, the destination node replies with a report listing all successfully received blocks. After receiving the report, the originator will retransmit missing blocks and discard those that were successfully received.

Since bundles may be buffer for long periods of time before being acknowledged, buffer saturation becomes highly common resulting in excessive latencies and bundle losses. This is aggravated by the fact that, in this context, reactive congestion management would be severely impacted by delayed control decisions. To resolve this problem, the authors propose a proactive strategy that takes advantage of virtual entities known as Decision Makers (DMs) implemented in each node. Each DM utilizes one of three existing MADM-based forwarding algorithms to perform optimal next-hop selection on a bundle per bundle basis.

The proposed algorithms are named after the three different decision-making approaches provided by MADM, namely:

1) *Simple Additive Weighting* (SAW) [85]: selects the best next-hop that minimizes the sum of all attributes of interest.
2) *Minimum Distance with Utopia Point* (MDUP) [86]: selects the best next-hop based on the knowledge of ideal alternatives ($i.e.$ utopia points) characterized by a time dependent coordinates of utopia attributes: the cost ($i.e.$ buffer occupancy) and the benefit ($i.e.$ available bandwidth). In this regard, next hop selection is based on the dual objectives of minimizing cost while maximizing benefit.
3) *Technique for Order Preference by Similarity to Ideal Solution* (TOPSIS) [85], [87]: In addition to MDUP's utopia points, TOPSIS also accounts for the worst alternatives ($i.e.$ nadir points) as they may be found to maximize the cost and minimize the benefit in light of the given attributes. The next hop selection is therefore based on the minimization of the Similarity to Positive-Ideal Solution. The reader may consult [81], [85] and [87] for more details about such solutions.

The adopted attributes are:

1) *Bundle Buffer Occupancy* (BBO): Ratio between the number of bundles stored in the buffer and the buffer size.
2) *Average Bandwidth* (AB): Link capacity between a node and its selected neighbor in (bits/sec).
3) *Transmission Time* (TT): Ratio of the bundle size (bits) to the average bandwidth (bits/sec).

Obviously, combining the strategies and the attributes together, results in a pool of six alternative congestion resolution schemes: SAW-BBO-AB, SAW-BBO-TT, MDUP-BBO-AB, MDUP-BBO-TT, TOPSIS-BBO-AB and TOPSIS-BBO-TT. Mono-attribute schemes such as SAW-BBO and SAW-TT were also considered for comparison purposes in a wide set of extensive simulations at:

1) Microscopic ($i.e.$ node) level, where metrics such as Bundle Buffer Queue Length (BBQL) and Bundle Buffer Filling Rate (BBFR) were used as performance indicators.
2) Macroscopic ($i.e.$ network) level where metrics such as Bundle Loss Rate (BLR) and Data Delivery Time (DDT) where used for performance evaluation.

One of the weaknesses that one can identify in this respect is the following. For a particular bundle, once the DM of a node decides upon a best next hop, this selection remains fixed throughout the transmission of all the bundle's blocks. For large enough bundles, the given attribute values may vary during transmission causing the current selection to become non-optimal. In addition, the requirement of attribute re-evaluation before the forwarding of individual bundles is time consuming and inefficient since it involves interactiveness and thus results in poor bandwidth utilization.

D. Vector-based Forwarding Protocols

The work in [88], propose a Flooding-based Vector Routing (FVR) protocol. FVR aims at reducing the number of message duplications in the network while achieving an acceptable performance in terms of delivery ratio and delivery delays. In FVR each node periodically extracts its location coordinates every Δt seconds and stores them. Given its coordinates (x_t, y_t) a time instant t, and those $(x_{\Delta t}, y_{\Delta t})$ at an instant $t - \Delta t$, it computes a current vector V_{cur} and records it. In order to accurately predict the movement direction, FVR computes the weighted moving average of the nodes ($i.e.$ their

trajectories) as: $V_t = \beta V_t + (1-\beta)V_t - \Delta t$ where β is a positive constant less than 1. As nodes encounter each other, they exchange their vectors that implicitly encapsulate information about their direction and velocity. Based on such information each node then makes a decision on which of its currently available neighbors may act as a next-hop for a given message and how many message copies should be replicated to each of those next-hops. Particularly, FVR favors neighbors moving in an orthogonal direction to the movement of the deciding node. A larger number of message replicas are forwarded to such nodes as they have better chances in meeting the destination. Nodes moving in the opposite direction have the possibility of traversing the same trajectory that the deciding node traversed earlier hence a minimal number of message copies are forwarded to such nodes. Nodes moving in the same direction of a current node, and with the same speed will not obtain any message replica because they are not going to reach the destination before that node. Forwarding a certain number of message replicas to nodes moving in the same direction of the current deciding node is possible though only if those nodes have a higher velocity and are, as such, most likely going to reach the destination first. In this case FVR relies on the velocities of the nodes in order to decide how many message copies to forward.

In an extension to their work in [88], the authors propose in [89] a History-based Vector Routing (HVR) where each node in the network creates and manages its own location vector history. A node also keeps a record of the vector information history of all its current neighbors as well as all other nodes it has previously encountered. In this way, each node sends to each neighbor the vector information it has. It also receives a similar chunk of information from each neighbor. As such, every node will possess a database that contains information about the location of all its neighboring nodes. This database is updated upon the occurrence of new encounters. Each of these databases will therefore serve to perform more efficient forwarding strategies since, for any particular bundle, the area where its ultimate destination is possibly located may now be predicted. For this purpose, given a bundle B held by a node N and destined ultimately to a destination node D, the authors defined the *rendez-vous probability* as the probability of a node M encountered by N meets with D.

Whenever a contact opportunity involves more than one neighboring node, the bundle B will be forwarded to the neighbor with the highest rendez-vous probability. In view of this, it is clear that, unlike other history-based protocols, HVR accounts for node mobility. Finally, in the case where HVR cannot predict the area of location of a destination, it will just act as FVR where replication occurs by only considering vector difference.

In both [88] and [89] extensive simulations are performed to evaluate the performance of FVR and HVR and prove their effectiveness compared to simple flooding and probabilistic routing. Nevertheless, this information exchange among encountering nodes, in our opinion, negatively impacts the entire design. Optimistically enough, we can say that this impact is much less in FVR than in HVR. The reason is that the amount of exchanged information is small ($i.e.$ the velocity and a simple vector computed a priori implicitly indicating the direction, only for a single node) and can be performed fast enough where bandwidth consumption becomes quite negligible. This condition holds only in sparse networks. However, as networks become denser, the encounter frequencies increase and the number of encountered nodes per opportunity also increases. Given that such exchanges are to be performed with all encountered nodes before data transmission can occur to the proper node, this directly implies that overhead increases and hence inefficiency in bandwidth utilization becomes more and more significant. This becomes even worse in HVR since the exchanged information covers all encountered nodes and it is clear that we will have scalability problems as networks become denser. This is not to mention that location vector information databases will become larger and larger utilizing valuable storage spaces that can be used otherwise for storing the more valuable data bundles. On top of that, a very important point comes to our attention in the case of HVR: even if a node possesses location information about a particular bundle's destination, in the context of moderate to rapid node displacement, there may often be cases where the destination's location information stored at the transmitting node becomes invalid either before the bundle's transmission or while the bundle is in transit. This is especially true whenever the destination is moving farther away from the transmitting node and its neighbors. In this case, even if different encounters occur among those latter, no information update about the destination in question will occur and false forwarding will take place possibly more than once.

E. Delegation Forwarding

The excess traffic caused by Epidemic Routing (ER) in DTNs has always been one of researchers' major concerns. Indeed, the large number of bundle duplications needed to achieve reliable delivery with minimal latencies is one of the major limitations of ER. To cope with such a limitation while preserving ER's merits, a simple yet powerful scheme known as Delegation Forwarding (DF) was introduced in the open literature [90]. It is basically based on assigning quality and level values to each and 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 replicas, etc.). At an initial stage, the level of a node is set to be equal to its quality and is used to measure the ability of a node in encountering other higher quality nodes in the network. Upon the encounter of two nodes, forwarding from one node to the other occurs only if that latter has a higher quality 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.

The authors in [91] aim at improving DF. They present a Probability Delegation Forwarding (PDF) scheme: For any arbitrary bundle B, upon the random encounter of its holder

with a higher quality node, N, a copy of B is forwarded to N with a probability $p \in [0;1]$ if and only if N has no such copy. Appropriate forwarding node level updates follow as in DF. With a probability $1-p$, B is retained in the forwarder's buffer with no modification of the level of this latter. The same process is repeated again and again upon the occurrence of encounters involving any one of B's copy holders with a higher quality node that has no such copy until B is finally delivered to its destination. This has the advantage of further reducing the number of bundle replicas. Intuitively, whenever $p = 1$, PDF reduces back to DF. The authors perform mathematical analysis to theoretically show that PDF's induced traffic is smaller than DF yet, they prove by simulations that it can still achieve the same delivery ratio as DF if p is not too small.

ER logically and clearly achieves the highest delivery ratio with the least delays but at the expense of excessive duplications. Thus, the farther away we move from nave random duplication the more traffic is expected to be reduced. However, this comes at the expense of longer delays and smaller delivery ratio ($i.e.$ an overall performance decrease). In simple ER, it is only those highly qualified nodes ($i.e.$ those nodes that are characterized by the highest encounter frequency) that are able to first deliver bundles as rapidly as possible. Moreover, in the context of ER, everybody is equally likely to receive bundles and be promoted. It is thus the low quality nodes with their very low encounter probability with bundle destinations that, on the overall, make this slight improvement of ER over DF. This is especially true since, in the latter case of DF, no bundle forwarding is made to those low quality nodes. However this performance decrease of DF is quite small and therefore is acceptable. In PDF the performance decreases further as compared to ER and surely is slightly worse than DF but is still acceptable.

It is important to note that when considering latencies, the authors noticed a significant gap between DF and PDF. They do not really elaborate on the cause of this gap. This is why we find it important to give the reader a feel for why such a gap exists. The reason is the following: when a node, say N, encounters another node M with a lower quality, this does not mean that M is not a good enough next hop. On the contrary, chances are that M might be the best next hop that N may ever encounter. By not forwarding a bundle to M, N risks to uselessly increase the delivery delay experienced by the bundle. On the overall, the simulations that the authors performed clearly show that this risk is high and hence the gap. In an attempt to bridge this gap, the authors define a quality threshold TH that is used to control the bundle exchange as follows. Upon the encounter of two nodes, if the receiving node's quality is above TH, a forwarding will occur with probability $p = 1$. Otherwise, the PDF approach takes over. This new strategy is referred to as Threshold Probability Delegation Forwarding (TPDF). It clearly enables a bundle to be delivered sooner to its destination. However, TH is quite critical when it comes to the traffic-latency tradeoff and should therefore be handled with care.

As a final note, the work in [91] is quite attractive, especially that the simulations accounted for relatively dense networks. Nevertheless, one may question the origin of this probability

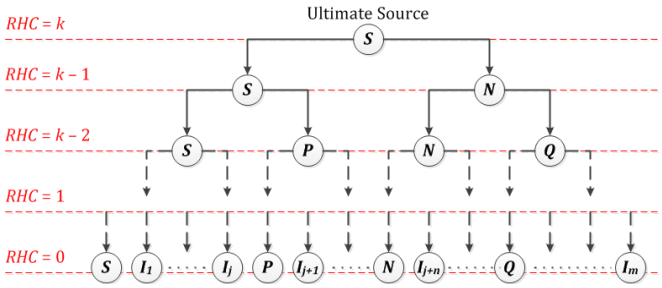

Fig. 19. Hop-Count Tree illustration.

p along with the factors involved in its computation. It seems that p is an administered value rather than an automatically computed one. As such, one possible future direction for the work in [91] would be to define an automated mechanism that computes p and let the protocol adapt accordingly.

F. Probabilistic Forwarding

Another strategy for cost limitation is presented in [92]. The Optimal Probabilistic Forwarding (OPF) is a hop-count limited probabilistic forwarding protocol that maximizes the expected delivery rate subject to a specific number of replicas for each bundle. The technique employed by OPF is somewhat similar to the Spray-and-Wait[9] (SW) technique found in [93]. In Limited Hop-Count Forwarding (LHCF) strategies, a *residual hop counter* (RHC) is used to limit bundle hopping. RHC is decremented on each hop. When RHC reaches zero, the associated bundle is discarded. For example, when a copy of a bundle B held by a node N with an $RHC_{B,N} = r$ is forwarded to another node M, then $RHC_{B,N} = RHC_{B,M} = r-1$. Thus, if ultimately at a source S of B the initial $RHC_{B,S} = k$, then B can be replicated at most $2^{k+1}-1$ times including the copy delivered to the destination as illustrated in Figure 19. In Figure 19, an ultimate source S has a bundle B with an associated RHC = k. Upon having a copy of B forwarded to a node N encountered by S, the RHC of B's copies at S and N are reduced by 1. However note that the RHC values at nodes S and N after being set to $k-1$ become independent in the sense that if either S or N forwards further, the RHC of the other will not be affected.

The operation of OPF is fundamentally based on two major components:

1) Network knowledge, including long-term regularity[10] of nodal mobility and total knowledge of nodal inter-meeting times.
2) Comprehensive[11] and dynamic delivery probability P_{i,d,k,T_r} of a bundle copy i destined to a destination d, coupling the residual hop count k and residual bundle lifetime T_r metrics for the purpose of optimizing performance.

The authors have recognized the major drawbacks of the probabilistic and delegation forwarding schemes that we extensively discussed earlier. In addition, they highlight the

[9]Each bundle, B, is associated with a limited number of copies, say C. Upon forwarding a copy of B, C is evenly redistributed among both copies.
[10]Nodal mean inter-encounter times can be predicted from history.
[11]Reflects both direct and indirect ($i.e.$ through intermediate nodes) delivery of bundle replicas.

importance of hop-counting through the fact that an encountered node may be a bad one-hop forwarder ($i.e.$ has a low direct delivery probability) but can be an outstanding two-hop forwarder ($i.e.$ frequently encounters a node with high direct delivery probability). They also stress the importance of residual lifetime as a factor controlling a bundle's eligibility for further forwarding. Under these conditions, whenever a forwarding decision is to be made, it will be probabilistic in nature and would follow the optimal stopping rule[12] described below:

Consider a node N, holding a copy c_B of bundle B with RHC equal to k. N encounters another node M. Forwarding a copy of c_B from N to M is consistently interpreted as replacing c_B at N by two new copies $c_{B,1}$ and $c_{B,2}$ respectively at N and M with equal RCH values of $k-1$. Such a forwarding event should only occur if and only if the intended copy replacement increases the probability of delivery. If no forwarding occurs, RHC of c_B will retain its value, k, but the residual lifetime will be decremented.

In the above rule, the authors have implicitly considered unicast forwarding in a sparse DTN. In addition, the time axis is subdivided into mini-slots and it is assumed that only a single forwarding may occur per time slot. The authors further assume that full routing knowledge is achieved through information exchange upon nodal encounters or global periodical updates. Such an assumption is later relaxed by assuming k-hop partial knowledge. It is clear that when no routing information is available, the delivery probability P_{i,d,k,T_r} on which OPF relies will be equal to 0 and thus OPF reduces to a simple Spray-and-Wait.

Using the same techniques, the authors developed also a Probabilistic Ticket-based Forwarding (PTF) algorithm where a bundle copy is associated with $L(L>1)$ logical tickets that are redistributed upon the occurrence of forwarding to the two resulting copies. Without loss of generality, if L_1 and L_2 were the number of tickets allocated to the two obtained copies, it is required that $L_1+L_2=L$. As much as it is possible for L_1 to be equal to $L_2=L/2$, it is also possible to have $L_1 \neq L_2$. The values of L_1 and L_2 are chosen in such a way so as to maximize the delivery probability. Similarly, if the forwarding will not result in an increase in that latter, then it will simply not occur but the RHC will still be decremented.

The authors further account for the broadcasting nature of wireless communications and realize that a single forwarding may result in multiple copies being distributed to the nodes available in an encounter opportunity. In view of this, they developed Broadcast Probabilistic Forwarding (BPF), a generalized ticket-based probabilistic forwarding scheme that operates as follows. Upon the occurrence of a forwarding event, tickets are divided among all the copies received by the entire set of available nodes using the same logic as in PTF. Again if no improvement of delivery probability is realized by a forwarding event, then forwarding will not take place.

[12]The details of stopping rules are outside the scope of this manuscript. Enough is explained in [92] and is left out to the reader.

G. Load Balancing-based Forwarding

The majority of the existing DTN forwarding algorithms adopt various heuristics for next hop selection aiming at achieving high throughput and efficiency. However, when such heuristics are applied to social-like network scenarios, they force those protocols to direct the majority of the traffic to a relatively small subset of reputable nodes. Consider for instance the Similarity-Betweeness (SimBet) algorithm [94] that relies on the combination of a decentralized version of egocentric centrality[13] and the probability of future nodal cooperation. In the case of SimBet, the top 10% of the nodes handle almost 54% of all the forwards and 85% of all the handovers. This results in an unbearable and unfair network load distribution that rapidly exhausts constrained resources ($e.g.$ storage space, battery, etc.) in highly popular mobile devices ($e.g.$ PDAs, cellular phones, etc). In addition, given that a small number of users handle a huge amount of traffic, the system is no more robust to random failures as the collapse of a single node can yield significant losses. Systems such as the Internet, road networks and airline traffic complex networks display a fat-tail connectivity distribution where some nodes handle a large number of connections while others have only very few. In such networks, the highly connected nodes handle the majority of the traffic, hence the unfair load distributions. However, this problem can be solved by simply equipping tied up nodes with extra resources and capabilities ($e.g.$ setup of high-end switches with high processing speeds and storage space, construction of more highways and roads, addition of extra airport terminals, etc). Such a solution is not applicable to DTNs where each node may belong to a different administrative domain ($i.e.$ different individuals that are not willing or do not have the budget for upgrading). Hence, failures and losses can be caused by: a) nodal resource depletion, b) attacks targeting reputable nodes, and c) cost-utility mismatches of reputable nodes forcing them not to cooperate. This motivated the authors in [94] to investigate fairness and load balancing in DTN forwarding schemes. Particularly, they target these two goals through the development of the so-called FairRoute that relies on two major metrics inspired from social science.

The first metric is the *Perceived Interaction Strength* (PIS), which is an indicator of the likelihood that a contact be maintained over time. The strength of an interaction between two nodes is measured by frequency of encounters. It is evaluated through both long and short term analysis. The authors account for these variations and define an aggregate PIS metric that is biased towards the more frequent long-term interactions and that penalizes fraudulent bursty activities. Social science is out of the scope of this manuscript. However, it is important to attract the reader's attention to an important point. For this purpose we made up the following scenario:

Sam would like to send a *Birthday* party invitation letter to *David*. However, on a holiday, all post offices are closed. While on his way to the park, Sam encounters both David's mother, misses *Miller* and her childhood friend *Nancy*. Both of them know David. Sam decides to hand-in the letter to either of them since both are able to deliver it to David. He thinks,

[13]Quantification of a node's self importance in a network.

after their promenade at the park, Nancy is highly likely to go back to her own place and will not probably see David today. After all she is not as close to David as his mother. This is why Sam decides to send the letter with Mrs. Miller. Truly, David calls Sam that night to confirm his presence.

In the above scenario, even though it was not explicitly stated, it is Sam's human intelligence that determined the stronger bond between David and his mother and thus achieved a successful delivery of his letter later that day. Such intelligence is not present among electronic devices. This is why, analogously, upon the encounter of a node S holding a bundle B ultimately destined to D with nodes M and N, PIS information on D is exchanged. Armed with such information, FairRoute comes into play computing both M and N's perceived utilities with respect to the delivery of a message to D. A decision is then made to forward B to the node with the larger utility value.

The above decision making mechanism is nonetheless not enough to perform load balancing since traffic will end up being directed to nodes with higher connectivity. To address this issue, a second metric Assortativity is used to limit bundle exchanges to those nodes having equal or higher "social" status. In this context, the authors define a node's social status as its queue length and claim that it is an indicator of its reputation ($i.e.$ how frequently this node is chosen to forward messages). Obviously, for the receiving node, accepting a bundle has a certain cost ($e.g.$ storage until delivery) thus incoming bundles will only be admitted if they are received from nodes with equal or higher status. Clearly, high status nodes will have the privilege of faster message forwarding whereas low status nodes will have to find other alternative paths.

By using assortativity, the authors are attempting to create a level/quality forwarding mechanism such as the one used in DF. Since nodes only forward to the ones having equal or higher status, this is quite equivalent to the quality measure defined in [90] and based on which forwarding occurs. In the context of FairRoute, the social status of a node is however upgraded automatically when it starts admitting more bundles. Building on this observation, the following problems may arise:

1) Since higher status nodes accept bundles only from nodes with equivalent or higher status, lower status nodes have extremely small chances to upgrade their status. As a result, such nodes might be stuck at low status for long intervals. Moreover, if the bundles stored at a low status node expire before status upgrade takes place, they will get discarded. Consequently, the node will be relegated to an even lower status.
2) We already know from earlier discussions that a DTN's node is given the freedom to act at its own incentive. Therefore, a node can be selfish enough or lazy (if we want to be more innocent) and not forward bundles further causing them to accumulate in its buffer. As a result, it will become a high status node. Such a node can start to act maliciously by discarding some of the bundles it has irrespective of their lifetime while striving to preserve its high status. This inevitably degrades the network's performance. Ultimately, since tracing and troubleshooting is almost impossible in most DTN scenarios, such nodes might get away with such a malicious behavior. This is a huge security issue that is left uncontrolled in FairRoute.

H. Encounter-based Forwarding

An observed characteristic of a subset of vehicular networks in disaster recovery scenarios is that some nodes ($e.g.$ those mounted over ambulances, police vehicles, fire fighters, etc.) tend to have an elevated level of encounters relative to other nodes. Therefore, future nodal encounter rates may be predicted from information relating to previous encounters. Based on this observation, the authors in [95] propose a DTN quota-based routing algorithm called Encounter Based Routing (EBR) aiming at achieving high delivery ratios while minimizing network overhead and resource utilization. In EBR, each node maintains two values, namely:

- Encounter Value (EV) that represents the exponential weighted moving average encounter rate associated with a given node.
- Current Window Counter (CWC) that reflects the number of encounters occurring during a particular time interval W.

Time is therefore divided into discrete intervals, each of length W. During an interval, CWC is incremented upon the occurrence of an encounter. At the end of each interval, EV is updated to account for both the present CWC and the past rate of encounters. Then, CWC is reset back to zero. It is worth noting that EV is a measure of a node's ability to deliver a bundle to its ultimate destination.

Upon the encounter between two nodes, EV values are exchanged and each node computes its own relative EV value where, by definition, the relative EV value of a node X with respect to a node Y is given by:

$$EV_r(X) = \frac{EV_X}{EV_X + EV_Y} \tag{1}$$

Let m_Y denote the number of copies of some bundle M that a node Y initially holds. When Y runs into another node X, Y computes the number of copies to be forwarded to X as follows: $c_X(m_Y) = m_Y \cdot EV_r(X)$. The number of M's replicas left at Y is then reduced to $m_Y - c_X(m_Y)$. Equation (1) places a tight constraint on the number of copies that a node is allowed to relay and as a result the node will be very rarely able to transfer all of its copies to the other nodes it encounters. This can be a limiting factor in the context of disaster recovery networks where resources ($e.g.$ power and buffer space) are very limited and sudden node destruction is highly probable.

In addition, the authors recognize the existence of the following security issue in EBR. A malicious node can practically convince any regular node to relay to it the majority of its messages by publishing high enough EV values. Once such messages are received, they may be completely dropped (i.e DoS attack) or partially dropped in order to slow down and decrease the chances of message delivery. The solution proposed by the authors for this problem is not quite efficient.

Fig. 20. Rollerblading activity in Paris, [96].

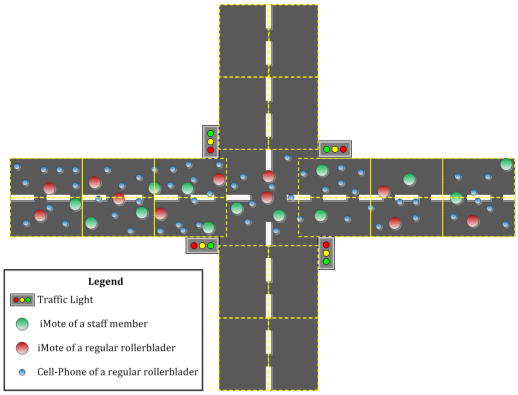

Fig. 21. RollerNet Experiment as illustrated in [95].

In fact, the authors require that each node present enough evidence that its published EV value is not forged by having each node retain a list of encounters each of which digitally signed by the corresponding previously encountered node. In this way, when a node X encounters a node Y, X would have to provide both its EV value and the digitally signed list of past encounters to Y. Armed with these different pieces of information, Y would be able to re-compute the EV value based on the information given in the list and then compare the computed value it to the one provided by X. If there is a match, then Y will confidently forward replicas to X.

The above security process is left optional. However, an obvious problem appears when a number of malicious nodes interact so often and help raise each other's EV values while still conforming to the security rules. Detecting such a situation is indeed extremely challenging.

I. Resource Allocation-based Forwarding

In [96], a utility driven Resource Allocation Protocol for Intentional DTN routing (RAPID) is presented to explicitly optimize a network administrator-specified metric. In the context of RAPID, two nodes communicate when in transmission range of each other where a sender replicates bundles to a receiver while retaining a copy of each. Bundles may be delivered from source to destination through direct contact or through intermediate nodes. In addition, the following assumptions are made. First, contacts are short. Second, bandwidth and storage capacity are limited. Last but not least, destinations are exceptionally assumed to have enough buffer space to hold all delivered bundles. In light of these assumptions, the authors investigate how to replicate bundles so as to optimize a specific routing metric.

RAPID discovers network resources via a control plane that helps nodes in acquiring complete information about network state. Each node exchanges such information ($e.g.$ number and location of replicas, average size of past transfers, delivery acknowledgements, etc.) with its neighboring nodes over an in-band control channel that utilizes a fraction of the available bandwidth. When combining exchanged information with a locally computed bundle marginal utility measure, RAPID makes a local bundle replica forwarding decision with justified resource utilization contributing to the optimization of a global metric. RAPID can be thus viewed as an extra greedy forwarding algorithm that replicates bundles in decreasing order of their marginal utilities. More specifically, RAPID has three major components:

1) Selection algorithm that determines which bundles to replicate.
2) Inference algorithm that estimates the utility of a bundle in terms of the specified metric.
3) In-band Control channel that propagates the metadata required by the inference algorithm.

Significant work has been made in order to verify and prove RAPID's good performance. First, it was deployed and experimented by means of a real vehicular DTN testbed. Second, node encounter traces and their corresponding available bandwidth were collected in order to design and perform extensive trace-driven simulations that compared RAPID's performance to that of existing protocols. To the best of our knowledge, RAPID is the first DTN routing protocol to be realistically implemented and tested.

J. Spray-and-Wait Forwarding in Pipelined DTNs

Pipelined Disruption-Tolerant Networks (PDTNs) are a particular subclass of DTNs characterized by a one-dimensional topology. A typical incarnation of PDTNS is the network made up of wireless nodes carried by rollerbladers travelling across a large area of a city. Rollerblading is quite a famous activity in France, particularly in Paris, as illustrated in Figure 20. Staff and other public safety forces usually assist rollerbladers throughout their tours. Each rollerblader has a delayed reaction to the movements of others and usually has to adapt his speed to slopes, red lights, obstacles and particularly those in front rolling at lower speeds as well.

The authors in [97] perform experiments in the context of the above scenario. Each staff member carries a Bluetooth enabled iMote. Each one of the regular rollerbladers may carry either a similar iMote or a cellular phone. iMotes and cellphones are used to log contacts between participants in the roller tour as depicted in Figure 20. The authors observed a particular characteristic of the so established network: the accordion phenomenon. Under this phenomenon, a small variation in the stability of any node in the system can highly impact the states of the other nodes. In the rollerblade activity, this translates into alternating compressions and expansions of

the rollerblading crowd. For example, if rollerbladers at the head of the group slow their speed, the others will get closer and closer to them before they finally slow down. This causes higher network density at the head first and then at the tail, hence, a higher nodal connectivity. The opposite occurs when the rollerbladers at the head roll faster.

Next, the authors investigate the effect that the accordion phenomenon on two forwarding strategies, namely, ER and SW. The performance measures that were to this end were the delivery delay and traffic overhead. In an attempt to capture the dynamics of the network connectivity under this phenomenon, the authors had to define three new metrics:

1) *Average Node Degree*: the number of contacts a node has during a given interval of time.
2) *Connected Components*: a periodical measure of the number of connected components and the size of a giant connected component.
3) *Average Delay*: the average delay required for a bundle to travel from a node to another given contact opportunities.

All the above three metrics were found to follow cyclic oscillations in or out of phase relative to each other. The relationship between the average degree and the delivery delay was studied theoretically in the context of the ER forwarding algorithm under the assumptions of infinite bandwidth, contention free medium and infinite buffer space. The authors showed that when nodes have limited connectivity, delay becomes independent of the size of the network. Under such conditions, the flooding process becomes slow. On the other hand, delay was observed to drop hyperbolically with increasing density.

Based on this study, the authors proposed two versions of the SW forwarding scheme that mitigate the accordion phenomenon effect:

1) *Oracle-based Spray-and-Wait* (O-SW): dynamically determines the most adequate number of copies to be sprayed depending on the current status of the network and given some desired performance metric ($e.g.$ a desired value of average delivery delays).
2) *Density-Aware Spray-and-Wait* (DA-SW): a distributed version of O-SW. DA-SW dynamically chooses the necessary number of copies to achieve a constant average delay based only on local information ($e.g.$ the current average degree of a node). This scheme primarily relies on the set of RollerNet experiment measurements preloaded into the node.

PDTNS is quite an important particular scenario of DTNs. To the best of our knowledge, the authors in [97] were the first to investigate the dynamics of such networks. Their work is an incentive for other researchers to allocate further efforts in such investigations. However, their assumption of infinite bandwidth, buffer size and no contention especially in such relatively dense network environments are unrealistic and hence can be major sources of results inaccuracies. In addition, it is quite clear that the authors rely significantly on the network topology and the traces they obtained. This dependence is solidly confirmed by the fact that those traces were preloaded on each node and as such served as a major component of their protocols. This is not an appropriate approach since such knowledge is often unavailable and surely is not worth propagating due to the high dynamics of the network.

K. Network Coding-based Forwarding

Routing a batch of messages in a DTN emerged recently as an interesting problem. The emergence of such a routing problem was driven mainly by the limited resources and transmission opportunities in DTNs as well as the long delivery delays. The authors in [99] exploit the principles of network coding[14] and its advantage of reducing the number of transmissions in a DTN context to combat network overhead. In particular, they proposed a Network Coding-based Epidemic Routing (NCER) protocol that augments ER with the network coding efficiency.

In NCER, nodes maintain coded messages in their buffers where a coded message c is a linear combination of K original source messages. Assume a given node N_1 happens to be holding n coded messages $c_1, c_2, ...c_n$ upon its encounter with another node N_2. N_1 would transmit to N_2 an encoded version of all those n coded messages ($i.e.$ a linear combination, l, of them). As N_2 receives l and its corresponding coding coefficients, it either stores it in its buffer if space is available or linearly combines it with already existing messages in its buffer. Finally, the destination, receives a coded message upon its encounter with any other relay node. Since the coding coefficients and the coded messages are known, then decoding the K original messages is similar to solving a linear system with K linear equations where the K original messages are the unknowns. The decoding process is successful only when K different coded messages are received or in other words the rank of the decoding matrix is K. Otherwise, the destination will have to wait until it receives more coded messages.

Efficient Network Coding-based Protocol (E-NCP) was then proposed in [100] as an extension to NCER that aims at both increasing its efficiency and reducing its incurred message delivery delay. In E-NCP, a source transmits K pseudo coded messages to L randomly chosen nodes as done in the case of Binary Spraying (BS) [101]. BS is used in this case because of its proven capability in achieving low message delivery delays. Therefore by simply tuning the parameter L referred to as the maximum spray counter, it is possible to achieve a trade-off between the number of relay transmissions and the message transmission delay. Through mathematical analysis, the authors were able to prove that L must be in the order of $\Theta(\log K)$ so that the performance of E-NCP would not degrade dramatically.

VI. Cooperative Delay-Tolerant Networks

A. Defining Cooperation in DTNs

The need to combat the stringent limitations imposed by extreme environments established the basis for the design of avant-garde communication techniques. These techniques are expected to be useful in improving the performance of

[14]Transmission of a coded message that consists of a linear combination of part or all the messages in a node's buffers.

DTNs. The majority of the works done in this respect studied the performance of DTN forwarding algorithms in scenarios that assume totally cooperative nodes. However, based on the DTN reference architecture [15], each DTN node may autonomously decide whether or not to accept custody transfers of incoming bundles from other nodes. Nodal cooperation is therefore not fully guaranteed and must not be taken for granted. The *degree of nodal cooperation* has, therefore, a considerable impact on the performance of DTN algorithms. That is, a low degree of cooperation may not only decrease the efficiency of such algorithms but also be the cause of their absolute failure. In this section, we shed the light on DTN node behaviors over and above mobility.

In order to capture the essence of a node's behavior in the DTN context, we believe that *cooperation* is the willingness and ability of a node to participate in a bundle's delivery process. Indeed, a node might not be self-centered, but forced not to cooperate due to resource limitations ($e.g.$ buffer space, power, etc).

In the last decade, peer-to-peer communication and Ad-Hoc Networks appeared as a fertile soil for the study of cooperative techniques where research went in the direction of investigating three main issues:

1) How cooperation affects the network's performance?
2) Non-cooperative node detection.
3) Design of protocols that forcefully impose node cooperation.

B. Discouraging Selfishness

To motivate nodes to cooperate, researchers in [102] and [103] devised some game theoretic punishment mechanisms. The authors in [104] and [105] came up with a more intricate procedure based on monitoring the individual behavior of nodes in the network. A per node reputation history is maintained and used to make forwarding decisions. Another credit-based variation of such strategies appeared in [106] where credits are awarded to those nodes that cooperate in the forwarding process and taken away from nodes that do not participate in the process.

The authors in [107] do not focus on how such punishment procedures are applied to DTN nodes. Instead they take a closer look at a node's behavior in an attempt to define a node's cooperative degree. When a node suffers resource limitations, it might either drop an incoming bundle with a so-called Type-I Cooperation Probability P_{drop} or accept the bundle with a Type-II Cooperation Probability $P_{forward}$. In this way, a node will purposely avoid some forwarding opportunities to save some resources ($e.g.$ power). The authors used $(1 - P_{drop})$ and $P_{forward}$ as quantitative metrics to characterize the sensitivity of three mobility-assisted DTN forwarding algorithms, namely: ER, Two-Hop Relay and Binary SW. Sensitivity evaluation of these protocols was done in light of the Fully Cooperative Equivalent (FCE) network.

In [111], the authors develop a selfish behavior discouraging strategy based on the principles of barter: two nodes A and B that happen to be in communication range would establish a connection and then exchange two sets D_A and D_B, containing descriptions of the bundles that each of them holds. The authors then differentiate between two bundle classes:

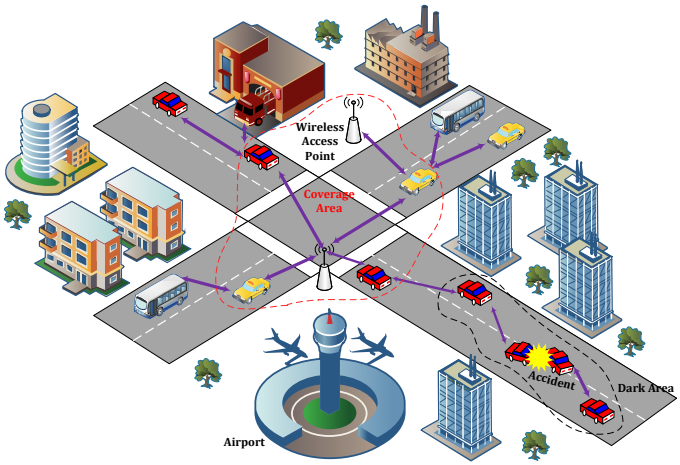

Fig. 22. Cooperative Vehicular Delay-Tolerant Network.

- Primary class: Bundle contents are of direct interest.
- Secondary class: Bundle contents are of no direct interest.

Bundles of both classes have the same barter value but different content values. Relative to a particular node, the contents of a primary bundle always have a greater value than any of the secondary bundles. Bundle values are independently assessed. Each node autonomously classifies bundles into one of the above classes based on their values. Therefore, some bundles may appear as secondary to a particular node but might be of primary value for another. A node may, therefore, be interested in downloading and storing secondary bundles for future exchanges of primary ones with other nodes. Message type, value and age are therefore the three metrics based on which the node decides whether or not to retrieve a message. Descriptions of secondary bundles with values less than a specific threshold, h, ($i.e.$ of no interest for future exchanges) are dropped. Also, descriptions of primary bundles with values below a download cost, c, ($i.e.$ their return is low relative to their cost) are dropped. Accordingly, D_A and D_B are updated and each node requests from the other the bundles whose descriptions are present in its set. For fairness purposes and following the aforementioned barter principle, both nodes should download an equal number of messages n equal to the cardinal of the smallest set.

The authors in [113] argue that when network nodes are controlled by rational entities ($e.g.$ people, organizations, etc), they tend to behave selfishly by only maximizing their own utility ($e.g.$ free-riding) irrespective of system-wide status. Such a behavior may drastically degrade other nodes' performance and often cause starvation and service denials. Motivated by this argument, the authors study the impact of such selfish node behaviors in DTNs. They generally formulate four variations of the DTN forwarding problem namely, the non-cooperative DTN (NC-DTN), the fully cooperative Synthetic Trace (FCST), the Haggle Trace[15] (HT) [114], and ZebraNet Trace (ZNT) [115], as four Linear Programs (LPs). Each of these LPs is solved independently.

Taking the probability of delivery within a given deadline as the performance metric, results show a critically impaired

[15]Compared to ZebraNet, Haggle is characterized by a higher network connectivity.

performance for NC-DTN. Therefore, the authors propose to robustly incentivize cooperation among DTN nodes through the development of a new Incentive-Aware Routing Protocol (IARP). They first identify two primary parameters in the context of incentive-aware DTN routing:

C. Cooperation in Vehicular DTNs

The work in [108], adapts a variation of the Cooperative Acknowledge Request (C-ARQ) strategy to VANETs where cars download delay tolerant information from Access Points (APs) on the road. This scenario is depicted in Figure 22. Areas lacking connectivity with at least one AP are referred to as dark areas. In these areas, cars rely on cooperation to exchange control and data bundles and request retransmissions of corrupted/lost bundles. The authors point out the considerable impact that vehicle positions have on cooperation opportunities. For example, they make it clear that vehicles moving in the vicinity of each other are subject to the same reception conditions and as such should not cooperate with one another. They also claim that their strategy will considerably decrease the number of APs a vehicle might need to communicate with. This can be effectively used to increase the APs' transmission rates while maintaining a relatively low packet loss rate.

The authors in [109] explain three operation phases of their proposed Vehicular DTN cooperative scheme and implement a real life 802.11-based prototype to test their hypothesis. However, this prototype has been implemented in the context of major simplifying assumptions such as: the negligence of the major impact that AP detection mechanisms ($i.e.$ association and authentication) can have on the overall performance of the network.

An extension to [109] appears in [110] where the authors propose a new Delayed Cooperative ARQ (DC-ARQ). In this scheme, cooperation between mobile nodes is delayed until they become out of the APs' coverage areas. This is basically the primary difference between DC-ARQ and C-ARQ where cooperation occurs on a bundle-by-bundle basis. DC-ARQ is not an end-to-end protocol and is therefore unable to guarantee the complete reception of a set of data ($e.g.$ a file). As the nodal density increases, it was shown through simulation and real-life experiments that DC-ARQ is able to achieve full bundle recovery. This is due to the fact that, in such scenarios, some nodes will always be in APs' communication range and thus able to receive bundles destined to other nodes. These nodes will highly cooperate when the delivery of missing or corrupted bundles is needed just as illustrated in Figure 22. This would not be possible in low nodal density scenarios where nodes might not be even found in APs' range and as such cannot receive bundles destined to other nodes. Under these circumstances, cooperation becomes unfeasible.

D. Storage-based DTN Cooperative Schemes

Opportunistic cooperative storage has been addressed in [110]. Authors of this work believed that, in scarce networks, opportunistic cooperative caching considerably improves information accessibility to mobile nodes. They observed that this technique is only limited to handle complete Application Data Units (ADUs), while some DTN scenarios, suffer short contacts and limited storage space. Bundle fragmentation is therefore required especially when coding-based forwarding protocols are to be used.

The authors first engaged in extending the notion of cooperative caching and storage to account for fragmentation. Using redundancy they increased all of the response probability, cache hit rate and reduced the response delays.

To favor nodal cooperation in this context, authors have developed a new DTN message-based content storage architecture. In addition, they modeled the erasure-code-based operation.

The authors supported their arguments with extensive simulations showing that adding redundancy either through erasure coding or flooding improves response probability. However, applying both simultaneously leads to no additional improvements. In addition, Caching at intermediate stages provides some performance gains but only if bundle lifetimes are long enough.

On a different scale, the authors showed that end-to-end response delays decrease with fragmentation. They further decrease if coding is applied. It is true that low response time is quite desirable. However, we recall that a reliable communication threshold should also be maintained. This is where redundancy comes into play. Nevertheless, too much redundancy increases queue length. This in turn decreases the likelihood of delivery within minimal delays, hence, the trade-off.

A recent work in [116] focuses on a class of distributed storage systems with time evolving contents. The authors aim at studying efficient time evolving file distribution methods. A popular example is a file containing weekly weather forecast updates. On a given day, any version of the file from the six last days may be useful for a user requesting forecast information of the next consecutive day. Recent file versions are the most accurate. In addition, having access to a given file version makes all its antecedents irrelevant. In a network of N mobile DTN nodes and a single source, at any given moment, an updated version of a time evolving file F is created by the source. In a non-cooperative setting, the source may deliver a copy of the updated version of F to a requesting node only during their encounter. However, in a cooperative environment where all nodes cooperate, any two nodes that encounter each other exchange file information and the node with the most updated copy of F transmits the file to the other node. In both settings, upon the reception of a more recent version of F, the receiving node discards the older version. Interestingly enough, each node in the network will hold at most one copy of F. Furthermore, a node is said to have an age $k \geq 1$, if the source has updated the file F an amount of $k-1$ times since its creation. A node has an age of 1 if it holds the most updated version of F. It has an age 0 if it does not hold any copy of F. In this context, the authors defined a *file management policy* as a set of rules specifying whether the source and a node, or two nodes, should communicate when they meet."

A policy is therefore said to be dynamic if the transmission decision depends on the node's age ($i.e.$ node's state). The

authors derive an optimal static file management policy and show that there exists an optimal threshold-based dynamic policy. Both policies have the objective of maximizing the system's utility given a power consumption constraint. Evaluation of both policy types are made in cooperative and non-cooperative settings. In a cooperative setting, however, any two nodes cooperate with probability b.

Finally, in order to stress on a major characteristic of DTNs where global network information is not available, the authors focus on non-cooperative settings and restrict to static ($i.e.$ local) policies assuming that global network knowledge are not available at the source ($i.e.$ N and encounter probabilities are unknown). Using stochastic processes theory they develop approximation algorithms that converge to the optimal static policy they have already derived.

E. Robust DTN Cooperative Forwarding Schemes

The work in [112] addresses robustness in cooperative DTN forwarding schemes. The authors propose a Cooperative Robust Forwarding scheme using Erasure coding (CORE).

Erasure coding consists of encoding one bundle with n blocks into m blocks ($m > n$). The original bundle can therefore be recovered from any n of the m blocks. However, the authors argue that the use of erasure coding alone is not sufficient to achieve message deliveries that are faster than the ones realized by typical duplication based schemes. A complementary parameter was introduced to reflect the capability of a relay node to deliver bundles to their intended destination.

CORE's basic operation revolves around this delivery capability parameter and is summarized as follows: As a message is generated, the source encodes the message into a large number, K, of smaller time stamped blocks that it holds until it encounters another relay node. Upon an encounter, nodes first exchange descriptive information about message blocks that each holds in its buffer as well as delivery capability related information. Then, based on the exchanged information, each node individually computes the delivery capability of the other in order to make the appropriate forwarding decisions. In cases of no buffer space shortage, nodes first drop expired message blocks. If buffer spaces are still not enough, the authors propose an innovative Cache Replacement Strategy (CSR) where nodes then drop messages that are less likely to be delivered than the messages about to be received.

- *Price of Anarchy* (PoA): Measures the effectiveness of an incentive mechanism in limiting the damage of selfish nodes.
- *Price of Incentive* (PoI): Measures the performance loss of incentivized cooperative nodes relative to the optimal performance of these nodes in the absence of any incentive mechanism.

Second, they set a very well defined forwarding objective to maximize traffic delivery within a given deadline. Therefore, the proposed IARP explicitly optimizes the mean link delay as a system-wide metric. IARP also keeps track of the variance of link delays to combat high link characteristics' variability. In addition, it employs Generosity-and-Contrition-based Tit-For-Tat (GC-TFT) where the new generosity and contrition parameters enable initial cooperation and suppress lengthy retaliations between any two neighboring nodes. Moreover, IARP is tailored in such a way to support large feedback delays and multi-hop paths.

VII. Open Research Problems

Delay and disruption tolerance is a novel emerging wireless networking paradigm that has not yet reached maturity. DTN Protocols are still in their infancy, large-scale DTN deployments do not exist at all, and small-scale deployments are still at their very early stages. These factors make real-life evaluation of present DTN modus operandi difficult. Indeed, rare are the protocols that were implemented, tested in real-life and proven to be free of lethal stealthy assumptions. But, the high costs and complexities of deploying real DTN testbeds ($e.g.$ deep space networks) leave us currently with no other choice but to evaluate the performance of DTNs through simulation studies built around many unrealistic assumptions.

This manuscript surveys a carefully selected set of recent works. Two other excellent surveys [10] and [11], provide insight into a wide variety of older schemes. Yet, various DTN challenges prevail as open research topics. In what follows, we discuss some of the DTN-related topics that lend themselves nicely to further investigation. We start by listing what, in our opinion, are the most essential problems to which attention must be directed first. We believe that solutions to those problems will capture the essence of DTNs and allow for interesting advancements in the field, powerful and efficient DTN protocol developments.

A. Essential Challenging Problems

1) Analytical Modeling and Performance evaluation of DTNs may be one of the most important research areas. DTN characteristics vary from one environment to another. Thus the development of a generalized DTN model is quite a challenging problem.
2) None of the routing protocols proposed in the DTN open literature specifies a clear-cut procedure for setting up paths between communicating nodes. The separation between the control plane ($i.e.$ determination of routes) and the forwarding (data) plane in the context of DTNs is clear. However, while significant efforts have been invested in handling forwarding issues, control has not yet gained a lot of attention.
3) Recall that DTN nodal contacts may be classified from highly deterministic to predictable all the way to absolutely unknown opportunistic. However, the less network information is available, the more the need for learning procedures increases. Such procedures are bandwidth consuming. The design of more intelligent, efficient and chattiness free network learning procedures is of particular interest.
4) When a bundle is received by its ultimate destination, its remaining replicas become useless. Instructing intermediate nodes to discard such copies requires additional resources. This is yet another form of the unresolved

issue relating to the tradeoff between efficiency and overhead.

5) Bundle security is still at its early stages. No security standards have been defined yet. We expect a lot of future work to be done in this direction.

What follows is a list of other persisting challenges that span the different DTN subsectors. None of the following problems has a priority over the others. They are listed arbitrarily with no particular order.

B. Other Persisting Problems

1) Erasure coding involves a lot of processing and hence requires more power. However, it was shown to improve the worst-case delay in [74]. This is particularly useful when applications require that bundles be delivered within a specific time interval. We expect more future investigations in this direction.
2) As a node becomes congested, incoming bundles may be dropped due to buffer overflow. This increases the dropping rate and adds network overhead as the forwarder inefficiently ends up consuming precious bandwidth to transmit of bundles that are dropped. An intelligent way to cope with this problem is to let a receiving node inform the forwarder of the probability p that inbound bundles are going to be dropped. Below a particular bandwidth occupancy threshold B, a receiving node always accepts incoming bundles. However when B is exceeded, p must increase. Analytical studies that determine an optimal value for B and that further explores the variability of p are future works of particular interest.
3) Due to short contacts and large inter-encounter intervals, some stored unexpired bundles may not have enough residual lifetimes for a contact to occur. In particular scenarios ($e.g.$ deep space) bundles may even expire while propagating to their intended receivers. Early discarding of those bundles may significantly reduce buffer occupancy and appears as an interesting congestion control mechanism.

VIII. Conclusions

The fundamental assumptions that lead to the birth of the Internet are no longer valid when considering a newly emerging era of intermittently connected wireless networks. Such networks operate in extreme environments characterized by challenging conditions. Nevertheless, regardless of all those stringent limitations, a wide variety of wireless network applications were still expected to be supported. This is why researchers proposed a new network prototype referred to in the open literature as a Delay- or Disruption-Tolerant Network (DTN).

There has been significant research advancement in DTNs. A large number of papers have been published. All of them aim at providing suitable solutions to different DTN problems in the context of some specific scenario. Our second goal has thus been to survey a valuable selection of eighty papers we believe are most relevant to the following DTN research concentration areas: routing, congestion and flow control, buffer management and cooperative strategies. The majority of these proposals were supported by custom-made simulations that show their advantages in the specific scenarios under which they were designed. We have observed a common performance evaluation practice in the majority of these works. The performance of a particular protocol that considers particular aspects under the context of a specific scenario is compared to those of other protocols that consider totally different aspects and that were devised for totally different scenarios. Such comparisons are not constructive. Hence, a unified evaluation framework is required. However, we are not really sure that such a framework can exist since to date all proposals are scenario specific. This is why we encourage putting more effort into implementing real-world testbeds and compare simulation results to real-life measurements.

IX. Must Read References

This section lists what we believe are must read references as they occur in their order of listing in our references section, namely: [2], [5], [6], [9], [10], [11], [12] through [15], [18], [42].

References

[1] J. F. Kurose and K. W. Ross, "Computer Networking: A Top Down Approach," 5th Edition, Addison-Wesley, Pearsons Education INC., One Lake Street, Upper Saddle River, NJ 07458, 2010.
[2] K. Fall, S. Farrell, "DTN: An Architectural Retrospective," IEEE J. Sel. Areas Commun., Vol. 26, No. 5, June 2008.
[3] M. Loubser, "Delay Tolerant Networking for Sensor Networks," SICS Technical Report, ISSN 1100-3154, January 2006.
[4] P. Basu and T. Little, "Networked Parking Spaces: Architecture and Applications," IEEE Vehicular Technology Conference (VTC), 2002.
[5] M. Demmer, "A Delay Tolerant Networking and System Architecture for Developing Regions," PhD. Dissertation, University of California, Berkeley, 2008.
[6] J. Heidemann, W. Ye, J. Wills, A. Syed and Y. Li, "Research challenges and applications for underwater sensor networking,"Proc. IEEE Wireless Communications and Networking Conference, 2006.
[7] M. Motani, V. Srinivasan and P. S. Nuggehalli, "PEOPLENET: Engineering a Wireless Virtual Social Network," in Proc. ACM/IEEE MobiCom, 2005.
[8] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass and J. Scott, "Impact of Human Mobility on The Design of Opportunistic Forwarding Algorithms," Proc. IEEE INFOCOM, 2006.
[9] Delay-Tolerant Networking Research Group, http://www.dtnrg.org/wiki.
[10] Z. Zhang, "Routing in Intermittently Connected Mobile Ad Hoc Networks and Delay Tolerant Networks: Overview and Challenges," IEEE Communications Surveys and Tutorials, Vol. 8, Issue No. 1, pp. 24-37, January 2006.
[11] 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.
[12] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, E. Travis and H. Weiss, "Interplanetary Internet (IPN): Architectural Definition," Available Online: http://www.ipnsig.org/reports/memo-ipnrg-arch-00.pdf.
[13] S. Farrell and V. Cahill, "Delay- and Disruption-Tolerant Networking," Artech House INC., Foreword p. ix, 2006.
[14] K. Fall, "A Delay-Tolerant Network Architecture for Challenged Internets," Intel Research Berkley, 2003.
[15] V. Cerf, S. Burleigh, A. Hook, L. Torgerson, R. Dust, K. Scott, K. Fall and H. Weiss, "Delay-Tolerant Networking Architecture," IETF Network Working Group, RFC4838, Informational, http://www.ietf.org/rfc/rfc4838.txt, April 2007.
[16] T. Spyropoulos, T. Turletti and K. Obraczka, "Routing in Delay Tolerant Networks Comprising Heterogeneous Node Populations," IEEE Trans. Mobile Computing, Volume 8, Issue 8, pp. 1132-1147, August 2009.

[17] I. Psaras, N. Wang and R. Tafazolli, "Six Years Since First DTN Papers: Is There A Clear Target?" First Extreme Workshop on Communications, Laponia, Sweeden, 2009.
[18] K. Scott, S. Burleigh, "Bundle Protocol Specification," IETF Network Working Group, RFC5050, http://www.ietf.org/rfc/rfc5050.txt, November 2007.
[19] S. Farrell and V. Cahill, "Evaluating LTP-T: A DTN-Friendly Transport Protocol," Proc. Third International Workshop on Satellite and Space Communications, 2007.
[20] I. Psaras, G. Papastergiou, V. Tsaoussidis and N. Peccia, "DS-TP: Deep-Space Transport Protocol," Proc. IEEE Aerospace Conference, Big Sky, Montana, United States of America, 2008.
[21] G. Papastergiou, I. Psaras and V. Tsaoussidis, "Deep-Space Transport Protocol: A Novel Transport Scheme for Space DTNs," ACM Journal of Computer Communications, Vol. 32, Issue No. 16, October 2009.
[22] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh and D. Rubenstein, "Energy-Efficient Computing For Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet," Proc. 10th International Conference on Architectural Support for Programming Languages and Operating Systems, San Jose, California, United States of America, December 2002.
[23] ZebraNet Project, Picture available online at: https://www.princeton.edu/eeb/gradinitiative/decisionmaking/zebranet.jpg
[24] A. S. Pentland, R. Fletcher and A. Hasson, "DakNet: Rethinking Connectivity in Developing Nations," IEEE Computer, Vol. 37, Issue No. 1, pp. 78-83, January 2004.
[25] A. Seth, D. Kroeker, M. Zaharia, S. Guo and S. Keshav, "Low-Cost Communication For Rural Internet Kiosks Using Mechanical Backhaul," Proc. ACM MobiCom, New York, N.Y., United States of America, pp. 334-345, 2006.
[26] M. J. Khabbaz, W. F. Fawaz and C. M. Assi, "Probabilistic Bundle Relaying Schemes In Two-Hop Vehicular Delay-Tolerant Networks," IEEE Commun. Lett., to appear, 2011.
[27] M. J. Khabbaz, W. F. Fawaz and C. M. Assi, "A Probabilistic Bundle Relay Strategy In Two-Hop Vehicular Delay-Tolerant Networks," Proc. IEEE International Conference on Communications (ICC) Wireless Networking Symposium, Koyoto, Japan, to appear, June 2011.
[28] R. Morris, J. Jannotti, F. Kaashoek, J. Li and D. Decouto, "CarNet: A Scalable Ad-Hoc Wireless Network System," Proc. Nineth European Workshop on ACM SIGOPS, New York, N.Y., United States of America, pp. 61-65, 2000.
[29] J.P. Singh and N. Bambos, "Wirless LAN Performance Under Varied Stress Conditions in Vehicular Traffic Scenarios," Proc. 56th IEEE Vehicular Technology Conference, Vol. 2, pp. 743-747, 2002.
[30] K. D. Lin and J. F. Chang, "Communications and Entertainment Onboard a High-Speed Public Transport System," IEEE Wireless Commun., Vol. 9, Issue No. 1, pp. 84-89, February 2002.
[31] L. Briesemeister and G. Hommel, "Role-Based Multicast In Highly Mobile But Sparsely Connected Ad-Hoc Networks," Proc. First ACM International Symposium On Mobile Ad Hoc Networking and Computing, 2000.
[32] C. D. Gavrilovich, "Broadband Communication On The Highways Of Tomorrow," IEEE Commun. Mag., Vol. 39, Issue No. 4, pp. 146-154, April 2001.
[33] I. Psaras, L. Mamatas and P. Mendes, "QoS Control in Next Generation IP Networks: An Experimental Analysis of Flow-Based and SLS-Based Mechanisms," Proc. Workshop on Networking in Public Transport, Waterloo, Canada, 2006.
[34] Haggle Project, available online at: http://www.haggleproject.org/.
[35] Million-People Project, available online at: http://www.amillionpeople.net/.
[36] SocialNets Project, available online at: http://www.social-nets.eu/.
[37] PeerSoN Project, available online at: http://www.peerson.net/.
[38] C. Huitema, "An Anycast Prefix for 6to4 Relay Routers," IETF Network Working Group, RFC3068, http://www.ietf.org/rfc/rfc3068.txt, June 2001.
[39] A. Acharya, B. R. Badrinath, T. Imielinski and J.C. Navas, "A WWW-Based Location Dependent Information Service For Mobile Clients," Rutgers University Technical Report, http://www.cs.ucsb.edu/~acha/courses/99/290i/papers/arup-location-dependent-info-service-for-mobile-clients.ps.gz, July 1995.
[40] Wikipedia The Free Encyclopedia, "MAC Address," Available Online: http://en.wikipedia.org/wiki/MAC_address, last update May 2010.
[41] K. Fall, S. Burleigh, A. Doria, J. Ott and D. Young, "The DTN URI Scheme," Network Working Group, Internet-Draft, Experimental, http://tools.ietf.org/html/draft-irtf-dtnrg-dtn-uri-scheme-00, March 2009.
[42] L. Wood, W. M. Eddy and P. Holliday, "A Bundle of Problems," IEEE Aerospace Conference, Big Sky, Montana, March 2009.
[43] J. Jackson, "The Interplanetary Internet," IEEE Spectrum, Volume 42, Issue No. 8, p. 30, August 2005.
[44] W. M. Eddy, "DTN Time Sync Issues," an e-mail to the IRTF dtn-interest mailing list and following discussions, April 2008.
[45] 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 Commun. Mag., pp. 1-9, June 2003.
[46] D. L. Mills, "Timekeeping in the Interplanetary Internet," University of Delaware, Available Online: http://www.ee.udel.edu/~mills/database/brief/ipin/ipin.pdf, August 2004.
[47] J. Postel, "Internet Control Message Protocol," IETF Network Working Group, RFC792, http://www.ietf.org/rfc/rfc792.txt, September 1981.
[48] J. Burgess, G. Bissias, M. Corner and B. Levine, "Surviving Attacks on Disruption-Tolerant Networks without Authentication, " Proc. ACM MobiHoc, pp. 61–70, September 2007.
[49] A. Davids, A. H. Fagg and B.N. Levine, "Wearable Computers as Packet Transport Mechanisms in Highly-Partitioned Ad-Hoc Networks," Proc. International Symposium on Wearable Computing, Zurich, October 2001.
[50] A. Lindgren, A. Doria and O. Schelen, "Probabilistic Routing in Intermittently Connected Networks," ACM SIGMOBILE Mobile Computing and Communications Review, Vol. 7, Issue No. 3, July 2003.
[51] M. Musolesi, S. Hailes and C. Mascolo, "Adaptive Routing for Intermittently Connected Mobile Ad Hoc Networks," Sixth IEEE International Symposium on Worl of Wireless Mobile and Multimedia Networks, pp. 183-189, 2005.
[52] P. Basu and S. Guha, "Effect of Limited Topology Knowledge on Opportunistic Forwarding in Ad Hoc Wireless Networks," Eigth International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WIOPT), Avignon, France, June 2010.
[53] S. Merugu, M. H. Ammar and E. W. Zegura, "Routing in Space and Time in Networks with Predictable Mobility," Georgia Institute of Technology, Technical Report, GIT-CC-04-7, March 2004.
[54] R. Handorean, C. Gill and G. C. Roman, "Accomodating Transient Connectivity in Ad Hoc and Mobile Settings," Second International Conference on Pervasive Computing, Vienna, Austria, pp. 305–322, April 2004.
[55] S. Jain, K. Fall and R. Patra, "Routing in a Delay Tolerant Network," Proc. ACM SIGCOMM, pp. 1–13, January 2004.
[56] A. Vahdat and D. Becker, "Epidemic Routing for Partially Connected Ad Hoc Networks," Duke University, Department of of Computer Science, Durham, NC, Technical Report CS-200006, April 2000.
[57] M. Grossglauser and D. Tse, "Mobility Increases the Capacity of Ad Hoc Wireless Networks," INFOCOMM, 2001.
[58] 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)," Proc. ACM MobiHoc, June 2003.
[59] D. Nain, N. Petigara and H. Balakrishnan, "Integrated Routing and Storage for Messaging Applications in Mobile Ad Hoc Networks," ACM Mobile Networks and Applications, Vol. 9, Issue No. 6, pp. 595-604, December 2004.
[60] F. Tchakountio and R. Ramanathan, "Tracking Highly Mobile Endpoints," ACM Workshop on Wireless Mobile Multimedia, Rome, Italy, July 2001.
[61] J. Su, A. Chin, A. Popivanova, A. Goel and E. De Lara, "User Mobility for Opportunistic Ad-Hoc Networking," Sixth IEEE Workshop on Mobile Computing Sysems and Applications, United Kingdom, December 2004.
[62] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh and D. Rubenstein, "Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet," Proc. ACM Architectural Support for Programming Languages and Operating Systems, San Jose, CA, October 2002.
[63] C. C. Shen, G. Borkar, S. Rajagopalan and C. Jaikaeo, "Interrogation-Based Relay Routing for Ad Hoc Satellite Networks," IEEE GLOBECOM, 2002.
[64] C. Becker and G. Schiele, "New Mechanisms for Routing in Ad Hoc Networks," Fourth Plenary Cabernet Workshop, Pisa, Italy, October 2001.
[65] Z. D. Chen, HT Kung and D. Vlah, "Ad Hoc Relay Wireless Networks Over Moving Vehicles on Highways," Proceeding of ACM MobiHoc, Long Beach, CA, pp. 247-250, 2001.
[66] R. C. Shah, S. Roy, S. Jain and W. Brunette, "Data MULEs: Modeling a Three-tier Architecture for Sparse Sensor Networks," IEEE Workshop on Sensor Network Protocols and Applications, May 2003.

[67] Q. Li and D. Rus, "Communication in Disconnected Ad Hoc Networks Using Message Relay," Science Direct Journal of Parallel and Distributed Computing, 63, pp. 75-86, January 2003.

[68] B. Burns, O. Brock and B. N. Levine, "MV Routing and Capacity Building in Disruption Tolerant Networks," IEEE INFOCOM, Miami, FL, March 2005.

[69] W. Zhao, M. H. Ammar and E. Zegura, "A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks,"Proc. ACM MobiHoc, pp. 187-198, 2004.

[70] W. Zhao, M. H. Ammar and E. Zegura, "Controlling the Mobility of Multiple Data Transport Ferries in a Delay-Tolerant Network," INFOCOM, 2005.

[71] I. Chatzigiannakis, S. Nikoletseas and P. Spirakis, "Analysis and Experimental Evaluation of an Innovative and Efficient Routing Protocol for Ad-Hoc Mobile Networks," SpringerLink Lecture Notes in Computer Science, Vol. 1982, pp. 99-111, 2001.

[72] I. Chatzigiannakis, S. Nikoletseas, N. Paspallis, P. Spirakis, C. Zaroliagis, "An Experimental Study of Basic Communication Protocols in Ad-Hoc Mobile Networks," SpringerLink Lecture Notes in Computer Science, Vol. 2141, pp. 159-169, January 2001.

[73] S. Jain, M. Demmer, R. Patra and K. Fall, "Using Redundancy to Cope with Failures in a Delay Tolerant Network," ACM SIGCOMM, August 2005.

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

[75] J. Widmer and J. Le Boudec, "Network Coding for Efficient Communication in Extreme Networks," ACM SIGCOMM Workshop on DTN, 2005.

[76] Y. Liao, K. Tan, Z. Zhang and L. Gao, "Combining Erasure-Coding and Relay Node Evaluation in Delay Tolerant Network Routing," Microsoft Technical Report, MR-TR-2006-05, January 2006.

[77] V. Conan, J. Leguay and T. Friedman, "Fixed Point Opportunistic Routing in Delay Tolerant Networks," IEEE J. Sel. Areas Commun., Vol. 26, Issue No. 5, June 2008.

[78] P. Basu and C. K. Chau, "Opportunistic Forwarding in Wireless Networks with Duty Cycling," Proc. ACM MobiCom Workshop on Challenged Networks, San Fransisco, C.A., September 2008.

[79] A. Jindal and K. Psounis, "Fundamental Mobility Properties For Realistic Performance Analysis of Intermittently Connected Mobile Networks," Proc. Pervasive Computing and Communications Workshop on ICMAN, 2007.

[80] T. Karagiannis, J. Y. Le Boudec and M. Vojnovic, "Power Law and Exponential Decay of Inter-Contact Times Between Mobile Devices," Proc. ACM MobiCom, 2007.

[81] I. Bisio, M. Marchese and T. De Cola, "Congestion Aware Routing Strategies for DTN-based Interplanetary Networks," Proc. IEEE Global Communications Conference (GLOBECOM), New Orleans, L.A., United States, November 2008.

[82] S. Burleigh, M. Ramadas and S. Farrell, "Licklider Transmission Protocol - Motivation," Network Working Group, Internet Engineering Task Force, RFC5325, http://www.ietf.org/rfc/rfc5325.txt, September 2008.

[83] M. Ramadas, S. Burleigh and S. Farrell, "Licklider Transmission Protocol - Specification," Network Working Group, Internet Engineering Task Force, RFC5326, http://www.ietf.org/rfc/rfc5326.txt, September 2008.

[84] S. Farrell, M. Ramadas and S. Burleigh, "Licklider Transmission Protocol - Security Extensions," Network Working Group, Internet Engineering Task Force, RFC5327, http://www.ietf.org/rfc/rfc5326.txt, September 2008.

[85] K. P. Yoon and C. L. Hwang, "Multi Attribute Decision Making An Introduction," Sage Publications INC., Thousand Oaks, C.A., United States, 1995.

[86] I. Bisio and M. Marchese, "Satellite Earth Station (SES) Selection Method for Satellite-based Sensor Network," IEEE Commun. Lett., Vol. 11, Issue No. 12, pp. 970-972, December 2007.

[87] F. Bari and V. Leung, "Multi-Attribute Network Selection by Iterative TOPSIS for Heterogeneous Wireless Access," Proc. IEEE Consumer Communications and Networking Conference, pp. 808-812, January 2007.

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

[89] H. Kang and D. Kim, "HVR: History-Based Vector Routing For Delay Tolerant Networks," Proc. IEEE Internation Conference on Computer Communications and Networks, San Fransisco, C.A., United States, pp. 1-6, August 2009.

[90] V. Erramilli, M. Crovella, A. Chaintreau and C. Diot, "Delegation Forwarding," Proc. ACM MobiHoc, pp. 251–259, May 2008.

[91] X. Chen, J. Shen, T. Groves and J. Wu, "Probability Delegation Forwarding in Delay Tolerant Networks," Proc. 18th IEEE International Conference on Computer Communications and Networks, San Francisco, C.A., United States, August 2009.

[92] C. Liu and J. Wu, "An Optimal Probabilistic Forwarding Protocol in Delay Tolerant Networks," Proc. ACM MobiHoc, New Orleans, Louisiana, United States, May 2009.

[93] T. Spyropoulos, K. Psounis and C. Raghavendra, "Spray and Wait: An Efficient Routing Scheme for Intermittently connected Mobile Networks," Proc. ACM SIGCOMM Workshop on Delay Tolerant Networks, Philadelphia, Pennsylvania, United States, August 2005.

[94] J. M. Pujol, A. L. Toledo and P. Rodriguez, "Fair Routing in Delay Tolerant Networks," Proc. IEEE INFOCOM, Rio De Janeiro, Brazil, April 2009.

[95] S. C. Nelson, M. Bakht and R. Kravets, "Encounter-Based Routing in DTNs," Proc. IEEE INFOCOM, Rio De Janeiro, Brazil, April 2009.

[96] A. Balasubramanian, B. N. Levine and A. Venkataramani, "DTN Routing as a Resource Allocation Problem," Proc. ACM SIGCOMM, Kyoto, Japan, August 2007.

[97] P. U. Toournoux, J. Leguay, F. Benbadis, V. Conan, M. Dias de Amorim and J. Whitbeck, "The Accordion Phenomenon: Analysis, Characterization, and Impact on DTN Routing," Proc. IEEE INFOCOM, Rio De Janeiro, Brazil, April 2009.

[98] D. Monniaux, Rollerblading tour in Paris, picture taken on the 12th of June 2006 and is available online at: http://commons.wikimedia.org/wiki/File:Paris_rollers_dsc03846.jpg.

[99] Y. Lin, B. Liang and B. Li, "Performance Modeling of Network Coding in Epidemic Routing," Proc. ACM MobiOpp, San Juan, Puerto Rico, United States, June 2007.

[100] Y. Lin, B. Li and B. Liang, "Efficient Network Coded Data Transmissions in Disruption Tolerant Networks," Proc. IEEE INFOCOMM, Phoenix, Arizona, United States, April 2008.

[101] T. Spyropoulos, K. Psounis and C. Raghavendra, "Efficient Routing in Intermittently Connected Mobile Networks: The Multiple-Copy Case," IEEE/ACM Trans. Netw., Vol. 16, Issue No. 1, February 2008.

[102] E. Altman, A. Kherani, P. Michiardi and R. Molva, "Non-Cooperative Forwarding in Ad-Hoc Networks," INRIA Technical Report No. RR-5116, Sofia, Antipolis, France, February 2004.

[103] V. Srinivasan, P. Nuggehalli, C. Chiasserini and R. Rao, "Cooperation in Wireless Ad-Hoc Networks," Proc. IEEE INFOCOMM, San Francisco, C.A, United States, April 2003.

[104] S. Buchegger and J. Le Boudec, "Performance Analysis of the CONFIDANT Protocol: Cooperation of Nodes—Fairness in Dynamic Ad- Hoc Networks," Proc. IEEE/ACM MobiHoc, Lausanne, C.H, June 2002.

[105] R. Molva and P. Michiardi, "Core: A Collaborative Reputation Mechanism to Enforce Node Cooperation in Mobile Ad-Hoc Networks," Proc. IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security, pp. 107-121, 2002.

[106] S. Zhong, Y. Yang and J. Chen, "SPRITE: A Simple, Cheat-Proof, Credit-Based System for Mobile Ad-Hoc Networks," Technical Report Yale/DCS/TR1235, Department of Computer Science, Yale University, 2002.

[107] A. Panagakis, A. Vaios and I. Stavrakakis, "On The Effects of Cooperation in DTNs," Proc. International Conference on Communication System Software and Middleware, Bangalore, India, January 2007.

[108] J. Morillo-Pozo, J. M. Barcelo-Ordinas, O. Trullols-Cruces and J. Garcia-Vidal, "Applying Cooperation for Delay-Tolerant Vehicular Networks," Forth EuroFGI Workshop on Wireless and Mobility, Barcelona, Spain, January 2008.

[109] J. Morillo-Pozo, J. M. Barcelo-Ordinas, O. Trullols-Cruces and J. Garcia-Vidal, "A Cooperative ARQ for Delay-Tolerant Vehicular Networks," Proc. IEEE International Conference on Distributed Computing Systems Workshops, pp. 192-197, June 2008.

[110] M. J. Pitkanen and J. Ott, "Redundancy and Distributed Caching in Mobile DTNs," Proc. ACM MobiArch, Kyoto, Japan, August 2007.

[111] L. Buttyan, L. Dora, M. Felegyhazi and I. Vajda, "Barter-based Cooperation in Delay-Tolerant Personal Wireless Networks," IEEE International Symposium on World of Wireless, Mobile and Multimedia Networks, pp. 1-6, June 2007.

[112] Y. Liao, Z. Zhang, B. Ryu and L. Gao, "Cooperative Robust Forwarding Scheme in DTNs using Erasure Coding," Proc. IEEE Military Communication Conference, pp. 1-7, October 2007.

[113] U. Shevade, H. H. Song, L. Qiu and Y. Zhang, "Incentive-Aware Routing in DTNs," Proc. IEEE International Conference on Network Protocols, pp. 238–247, October 2008.

[114] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot and A. Chaintreau, "CRAWDAD data set Cambridge/Haggle (v. 2006-09-15)," Available Online: http://crawdad.cs.dartmouth.edu/cambridge/haggle, September 2006.
[115] Y. Wang, P. Zhang, T. Liu, C. Sadler and M. Martonosi, "CRAWDAD data set Princeton/Zebranet (v. 2007-02-14)," Available Online: http://crawdad.cs.dartmouth.edu/princeton/zebranet, February 2007.
[116] E. Altman, P. Nain and J. C. Bermond, "Distributed Storage Management of Evolving Files in Delay Tolerant Ad-Hoc Networks," Proc. IEEE INFOCOMM, Rio De Janeiro, Brazil, pp. 1431-1439, April 2009.

Maurice J. Khabbaz received a B.E. and an M.Sc. in Computer Engineering with honors, both from the Lebanese American University (LAU) in Byblos, Lebanon, respectively in 2006 and 2008. He is presently a Ph.D. Candidate in Electrical Engineering at Concordia University, Canada. Between 2006 and 2008, he served as a Communications and Control Systems Laboratory and Computer Proficiency course Instructor as well as a Teaching and Research Assistant of Electrical and Computer Engineering at the Lebanese American University. He was appointed President of the LAU's School of Engineering and Architecture Alumni Chapter's Founding Committee and elected Executive Vice-President of this chapter in 2007. Presently he occupies a full-time researcher position in Electrical Engineering at Concordia University. His current research interests are in the areas of delay-/disruption tolerant networks, wireless mobile networks and queuing theory.

Chadi M. Assi is currently an Associate Professor with Concordia University (CIISE department), Montreal, Canada. He received his B.Eng. degree from the Lebanese University, Beirut, Lebanon, in 1997 and the Ph.D. degree from the Graduate Center, City University of New York, New York, in April 2003. Before joining Concordia University in August 2003 as an assistant professor, he was a visiting scientist for one year at Nokia Research Center, Boston, working on quality-of-service in passive optical access networks. Dr. Assi received the prestigious Mina Rees Dissertation Award from the City University of New York in August 2002 for his research on wavelength-division-multiplexing optical networks. He is on the Editorial Board of the IEEE Communications Surveys and Tutorials, serves as an Associate Editor for the IEEE Communications Letters and also an Associate Editor for Wiley's Wireless Communications and Mobile Computing. His current research interests are in the areas of optical networks, multi-hop wireless and ad hoc networks, and security. Dr. Assi is a senior member of the IEEE.

Wissam F. Fawaz received a B.E. in Computer Engineering from the Lebanese University in 2001. In 2002, he earned an M.Sc. degree in Network and Information technology from the University of Paris VI. Next, he received a Ph.D. degree in Network and Information Technology with excellent distinction from the University of Paris XIII in 2005. Since October 2006, he is an Assistant Professor of Electrical and Computer Engineering at the Lebanese American University. His current research interests are in the areas of next generation optical networks, survivable network design and queueing theory. Dr. Fawaz is the recipient of the French ministry of research and education scholarship for distinguished students in 2002 and of a Fulbright research award in 2008.