Delay Tolerant Networking - A Tutorial

Vinod Venkataraman
vinodv@cs.utexas.edu

Hrishikesh Bhatt Acharya
acharya@cs.utexas.edu

Harsh Shah
harsh@cs.utexas.edu

Simon Lam
lam@cs.utexas.edu

Department of Computer Sciences
The University of Texas at Austin
Austin, TX 78712-0233

ABSTRACT

Delay Tolerant Networking has been a hot topic of interest in networking since the start of the century, and has sparked a significant amount of research in the area, particularly in an age where the ultimate goal is to provide ubiquitous connectivity, even in regions previously considered inaccessible. Protocols and applications in popular use on the Internet are not readily applicable to such networks, that are characterized by long delays and inconsistent connectivity. In this paper, we summarize the wealth of literature in this field in the form of a concise, but comprehensive tutorial. The paper is designed to bring researchers new to the field with a general picture of the state of the art in this area, and motivate them to begin exploring problems in the field quickly.

1. INTRODUCTION

The Internet today widely operates on TCP/IP, thereby ensuring a standard set of protocols over which millions of devices worldwide operate. These protocols focus on providing end-to-end service (reliable or otherwise), while ensuring that variations in the underlying link-layer technology does not affect the basic working of the protocols. These protocols are largely characterized by paths that have low error rates, low delays and more-or-less continuous connectivity over time.

However, the protocols that are so widely used over the Internet may not be applicable to all kinds of networks, particularly those that operate under the constraints of high delays and losses. Examples of such networks include:

Terrestrial wireless networks: Such networks may connect mobile wireless devices, or devices in areas without much infrastructure to support continuous connectivity. Examples include networks in rural/remote areas, vehicular networks, etc. In rural areas, for instance, an ordinary passenger bus service could act as a store-and-forward switch, collecting data requests from various rural areas along its path, and delivering them to a bigger city with regular Internet connectivity.

Military networks: These networks operate under hostile conditions and involve large numbers of nodes (troops, vehicles, aircraft, satellites, sensors) where mobility is high and the environment may cause signals to be disconnected or jammed frequently. Such networks would also have varying priorities for different types of networks.

Atypical media networks: These include networks that operate in deep-space or underwater, and are usually characterized by high, but predictable latencies with periodic service, such as orbit of satellites, passage of ships, etc.

Such networks require protocols that take into consideration the specific communication needs of such networks, including link connectivity, delay, data rate asymmetry, addressing, reliability mechanisms, quality of service, etc. These parameters not only vary with those of the protocols used on the Internet, but also with each other. That is, such networks are typically mutually incompatible - *within* a network, communication is fluid, but *between* different networks, message exchange is not possible directly due to varying characteristics. Therefore, it is necessary to have a means to translate between such networks with mismatched delays.

A Delay-Tolerant Network (DTN) is a general-purpose overlay network that operates on top of varying regional networks, including the Internet. DTNs allow regional networks with varying delay characteristics to interoperate by providing mechanisms to translate between their respective network parameters. Therefore, the underlying protocols and technologies for these regional networks may differ considerably, but the flexibility of the DTN architecture allows them to be connected to each other.

In this paper, we summarize the wealth of literature in this field in the form of a concise, but comprehensive tutorial. We begin by describing the RFCs that treat the DTN architecture, and discussing the extensions and implementations of this architecture in Section 2. We then discuss various approaches to routing in DTNs in Section 3, and provide an analysis of the characteristics of these approaches. We

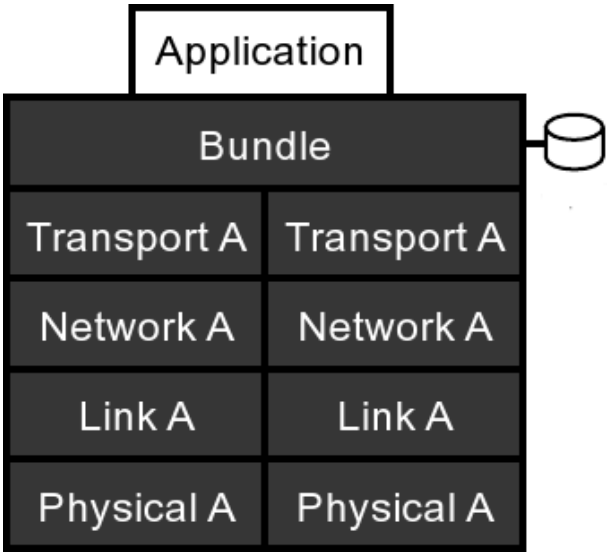

Figure 2: Logical view of a gateway

discuss security in Section 4, and analyze the various vulnerabilities in the architecture. We examine a case study in the form of a real implemented system - **KioskNet** - by discussing its specific architecture, implementation and applications in Section 5. Finally, we conclude in Section 6.

2. DTN ARCHITECTURE

Over the past few years, the Internet Research Task Force DTN Research Group [18] has performed considerable work in the area, and has published RFCs [15, 16] to describe the DTN architecture. In this section, we first describe the overlay architecture to allow interoperability of disparate challenged networks. This architecture was originally proposed by Kevin Fall in [9], and was later developed into RFC 4838 [15] by the DTN Research Group. The architecture is based on an abstraction of message switching, and is designed to operate as an overlay network above the protocol stacks in different networks and provide a store-and-forward gateway function between them. We discuss the key points of the DTN architecture in Section 2.1.

The DTN Research Group also drafted the Licklider Transmission Protocol (LTP) in [16] to provide retransmission-based reliability over links characterized by extremely long message round-trip times (RTTs) and frequent interruptions in connectivity. It was primarily drafted to support long-haul reliable transmission in interplanetary space where traditional protocols fail due to the long delays. We provide an overview of LTP in Section 2.2.

Further research has been undertaken to extend these RFCs to develop modified frameworks for DTNs. We examine some of these extensions, and give an overview of the available implementations of the DTN architectures in Section 2.3

2.1 The Bundle Layer

As described above, the bundle protocol is primarily designed to act as an overlay above various different types of networks. Message aggregates are formed by encapsulating application data into "bundles", and the routers that handle them are known as DTN gateways or bundle forwarders. Thus, the bundle layer stores and forwards entire bundles between nodes. The key features of the bundle layer architecture are described below:

Regions and Gateways: Figure 1 illustrates the concepts of regions and gatways in DTNs. A region is a specific type of network with specific protocols based on its communication requirements. In the above figure, four regions A, B, C and D are shown. Region B shows a DTN gateway (4) present on a passenger bus that cycles between DTN gateways (3) and (5). A gateway is basically a point of access between two regions that operate on possibly different architectures and protocol stacks. It has two logical halves, with each half in an adjacent region above their corresponding transport protocols. A logical view of a gateway is demonstrated in Figure 2. DTN gateways are responsible for storing messages in persistent storage when reliable delivery is required, and mapping between different transport protocols.

Naming and Addressing: In order to route DTN messages, *name tuples* are used, which consist of two variable length portions in the form **Region Name, Entity Name**. The first portion - the Region Name - is globally unique, and typically hierarchically structured. The second portion identifies a name resolvable within the specified region and need not be unique outside the region. For instance, the following name tuple would identify a particular entity on the Internet: {internet.icann.int, `"http://www.ietf.org/oview.html"`}

Postal-style delivery service: Since all the resources on DTNs are limited, it is necessary to impose a priority-based resource allocation and delivery mechanism. Similar to a postal service, the DTN architecture describes three relative priority classes (bulk, normal and expedited - conceptually equivalent to low, medium and high priority). It also supports several delivery options that may be selected by an application, including notifications of delivery, receipt and custody transfers.

Routing: The DTN architecture provides a framework for routing and forwarding. In this framework, a DTN network is defined by a multi-graph, where vertices may be connected by multiple edges. The contact between these vertices are classified as persistent, on-demand, scheduled, predicted and opportunistic, based on the network characteristics. The latter three contacts are most common among DTNs. The DTN architecture itself does not advocate any particular type of routing mechanism so as to provide maximum flexibity in regional designs. Routing is discussed in greater detail in Section 3.

Reliability and Custody Transfers: A custody transfer refers to the transfer of a bundle from one DTN node to the next and the corresponding passing of reliable delivery responsibility. The DTN architecture classifies nodes on the basis of their storage as Persistent and Non-Persistent, the former possessing sufficient bundle

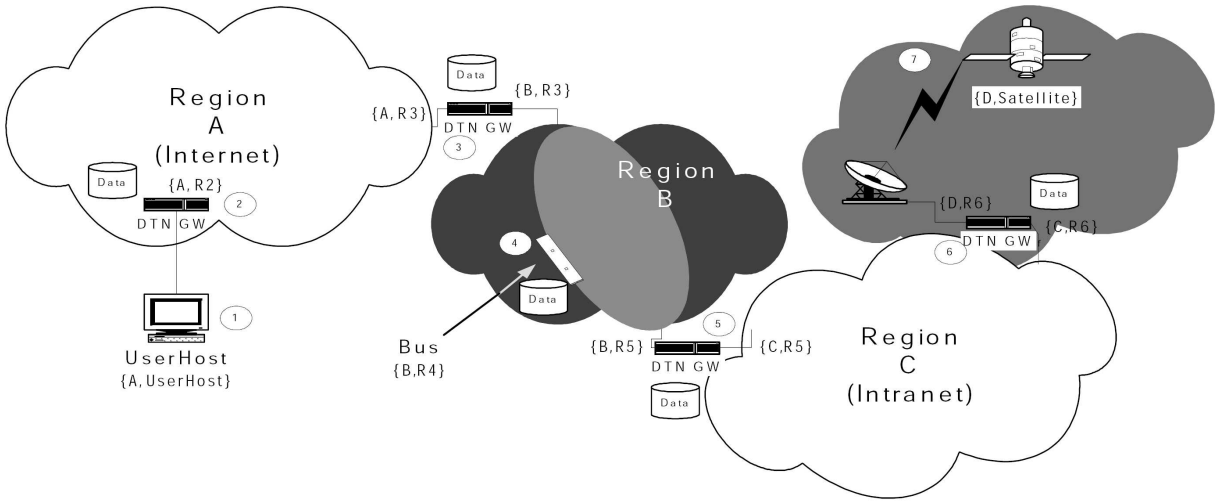

Figure 1: A sample collection of regional networks, reproduced from [9]

storage. This concept is very important in a network that has a high loss rate, and ensures that end-nodes, which may not have sufficient resources, are not burdened with holding bundles to maintain end-to-end reliability. Therefore, once a bundle has been custodially been transferred to a Persistent node, the source node need not maintain a copy of this data.

Convergence Layers: The features provided by the underlying layers of a DTN network, such as reliable delivery, connections (with indications of connection failure), flow control, congestion control, may vary considerably. Since the bundle forwarding protocol assumes reliable delivery from the underlying layers, it may be necessary to augment the stack with a specific convergence layer that ensures these features. For instance, in cases where reliable delivery is provided by an underlying transport, the corresponding convergence layer only needs to provide connection state management. The protocol stack for a typical DTN is shown in Figure 3.

Time Synchronization: The DTN architecture requires time synchronization among DTN nodes for purposes like bundle and fragment identification, routing with scheduled or predicted contacts and bundle expiration time computations. This is usually done by means of an external non-DTN protocol.

Flow Control and Congestion: Flow control in the DTN architecture refers to limiting the sending rate of a DTN node to the receiving rate of the next hop. Congestion control refers to the handling of contention for the persistent storage at a DTN gateway. The architecture states that flow control decisions must be made within the bundle layer itself, although it may use the mechanisms provided in the underlying transport layer to support it. Typically, these decisions are made when storage resources become scarce in a node. No particular method for congestion control is advocated by the RFC, and is cited as an open research problem.

Security: Security requirements for DTNs are required to restrict access to the scarce communication resources that are available. The goals of the security mechanisms are primarily to prevent unauthorized applications from utilizing the resources in the network, and to prevent authorized applications from accessing a higher class of service than they are entitled to. It is also necessary to identify and discard corrupted bundles, and detect compromised nodes. The suggested security mechanism utilizes hop-by-hop and end-to-end authentication and integrity mechanisms. The purpose of using both approaches is to be able to handle access control for data forwarding and storage separately from application-layer data integrity. Security is described in greater detail in Section 4.

2.2 The Licklider Transmission Protocol

LTP [16] is a retransmission protocol for delay-tolerant reliable communication between two points. It is designed to operate above the link layer, and serves as a convergence layer under the bundle layer for regional networks with extreme delay characteristics. A protocol stack schematic of LTP is illustrated in Figure 4. The key features of LTP are discussed below:

- LTP considers a block of data to be transmitted as two parts: a *red-part*, whose delivery must be assured by acknowledgment and retransmission as necessary, and a *green-part* whose delivery is attempted, but not assured. Thus, LTP can provide both TCP-like and UDP-like functionality concurrently on a single session.
- LTP splits a block into segments. The last segment of the red-part of the block is marked as the end of red-part (EORP), and as a checkpoint (identified by a unique checkpoint serial number) indicating that the receiver must issue a reception report upon receiving the segment. After the sender issues the EORP, a timer is activated so that the red-part can be automatically retransmitted if no response is received. When a

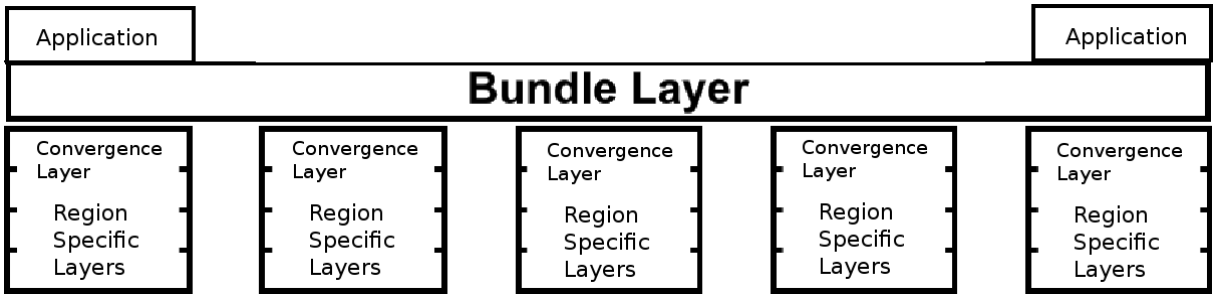

Figure 3: DTN Protocol Stack: Multiple convergence layers provide a common interface with the bundle layer

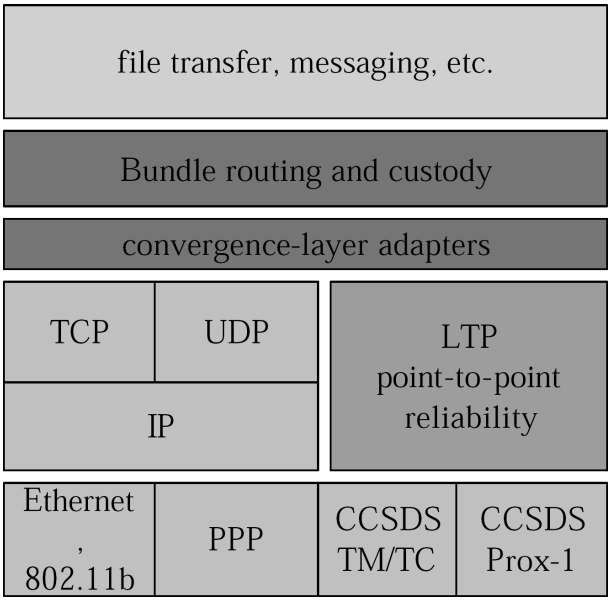

Figure 4: The Licklider Transmission Protocol

checkpoint is received, the receiver returns a report of cumulative reception of data from the previous checkpoint. The last segment of the block overall is marked as the end of block (EOB).

- Since LTP data flows are unidirectional, LTP's data acknowledgments - *reception reports* - cannot be piggybacked on data segments as in TCP. They are instead carried in a separate segment type.
- *Deferred transmission* is an important concept in LTP. The protocol provides link state cues to determine when it is and is not possible to transmit data. Therefore, LTP may be constantly generating outbound segments that may be queued for later transmission.
- LTP relies on accurate calculation of expected arrival times for report and acknowledgment segments in order to know when proactive retransmission is required. The lower bound on these arrival times is stricter than the upper bound, because if the expected time were early, it would result in a costly retransmission of data, whereas late expected times would only result in further delay in sending data.
- Retransmission occurs when the sender receives a report indicating one or more intervals of the red-part were corrupted or missing. In case of blocks with larger red-parts, it is possible to *accelerate* the retransmission by introducing multiple checkpoints within the red-part, instead of just at the EORP.
- The transmission session of an individual block may be *canceled* by either the sender or the receiver in response either to a request from the local client service instance or to an LTP operational failure.
- LTP includes two optional mechanisms in [17] to address security concerns. The security mechanisms and their motivations are discussed in Section 4.

The important strengths of LTP that make it an excellent protocol for networks where high delays are the norm is that it tolerates long link interruptions without data loss, and that it is designed to impose minimal overhead on low-capacity and asymmetric links.

2.3 Extensions and Implementations

Significant research has been undertaken to extend the architectures proposed by the DTN Research Group. These extensions typically include modifications to the original protocols to augment their functionality. In this section, we view two such proposals.

LTP-T: LTP-T is an extension to LTP designed by Farrell et al [10] that converts the original point-to-point protocol to a generic transport that is applicable over all links of a DTN, instead of just the long-haul links. Therefore, the protocol in essence converts the single hop LTP protocol to a multi-hop, potentially end-to-end protocol suitable for use as the transport protocol in the DTN. In certain cases, LTP-T works out to be more efficient than the bundle protocol, but is not as flexible as the latter. The main issues tackled by LTP-T are the provision of an API similar to sockets, the handling of storage congestion, and the use of the familiar addressing techniques of the Internet to provide ease of translation.

Adaptive Middleware: This work, by Petz et al [28] proposes an architecture where the assumption is that a networked device may exist in one type of region for a

period of time, and may later transfer to another region with a different architecture. This is in contrast with the original architecture's assumption that nodes in a region are restricted to that region and its corresponding network protocols. The adaptive middleware approach, therefore, proposes the use of a DTN service daemon that interfaces with the application, and a context aggregator that monitors the surroundings of the node and decides which network stack is suitable to the current environment. This middleware scheme allows the node to swap protocol stacks seamlessly as the node transitions from one region to another.

We now provide a brief discussion of existing implementations for DTNs. As DTNs are a relatively new concept, no commercial simulators / emulators currently support DTNs, although the research community has come up with several useful options. One of the earliest implementations by Demmer et al is detailed in [7]. This implementation, known as the **DTN Reference Implementation (DRI)** grew to become the most widely used implementation. Oliver et al conduct a series of benchmark tests on the DRI in [27]. Various other works develop specific implementations for the purposes of testing, but most are variants of the DRI. **Perfume** is an implementation of LTP that provides a socket API.

3. ROUTING

Traditional routing protocols operate under the assumptions of continuous connectivity, low delay and very low packet loss rate. However, as discussed in earlier sections, these assumptions are not valid in DTNs and therefore, protocols like distance vector routing and Optimized Link State Routing do not work properly. New routing protocols and system architectures are required to be developed for DTNs. Further, there are various types of DTNs based on their characteristics, and each such DTN requires its own specialized routing protocol. The DTN architecture [15, 9] treats various components of DTNs, but allows great flexibility for routing protocols in these networks based on their specific requirements.

The research community has responded with a vast amount of literature dealing with routing. In [37], Zhang provides an extensive survey of the state of the art in DTN routing protocols. In this section, we provide a summary of these protocols based on the type of DTN they are designed to service.

DTNs can be roughly classified into two categories: those with deterministic or predictable topologies and those with stochastic, time-evolving topologies. Each category has numerous approaches to routing, each with a specific set of assumptions about the network.

3.1 Deterministic Routing

In such protocols, the basic assumption is that the future movement of nodes and their connections are entirely known or predictable, and thus the network topology is known at the time of design of the architecture. Three approaches towards deterministic routing are outlined below:

Tree-based approach: In [13], Handorean et al propose algorithms for path selection based on available information on the motion of the hosts. Three cases are presented, the first of which assumes global knowledge of the motion and availability of hosts. In this case, a tree is built from the source host, and children nodes are added along with information on the time required for messages to reach them from the source. A path is ultimately chosen based on the best time to reach the required destination. The second case assumes that host information is initially unknown, but is eventually learned by allowing neighboring hosts to exchange characteristic information. The third case enhances the information learning in the second case by requiring that past information also be recorded.

Oracle-based approaches: In [19], Jain et al propose several routing algorithms based on the amount of knowledge about the network topology and traffic characteristics. They define four types of *knowledge oracles*, each representing a specific amount of knowledge about the network. Based on the availability of these oracles, various algorithms are proposed, such as a linear programming approach when all oracles are available, a modified Dijkstra approach if only certain oracles are present, etc.

Space-Time routing: In [24], Merugu et al assume that the network characteristics are known / predictable over a time interval **T**, rather than over the entire time horizon. The dynamics of the network is modeled as a space-time graph, and routing algorithms in this graph are developed using dynamic programming and shortest path algorithm.

3.2 Stochastic Routing

In such protocols, the base assumption is that the behavior of the network is random, and these protocols depend on decisions regarding where and when to forward messages. The simplest protocols choose to forward to any contacts within range, while other protocols base their decisions on history data, mobility patterns, or other information.

3.2.1 Epidemic-based approaches

In these approaches, it is assumed that absolutely no information is known about the movement of nodes and the topology of the network.

Epidemic Routing: In [36], Vahdat et al propose an epidemic routing scheme in which a node floods a message arriving on it to all other nodes in its neighborhood. Thus, all messages in the system are distributed to all nodes eventually. Of course, the protocol assumes that sufficient buffer space is available. Epidemic routing relies on message carriers to propogate the messages through the system through node mobility. This was possibly the earliest work in routing for intermittently connected networks.

Two-hop Forwarding: In [12], Grossglauser et al propose the opposite extreme scheme to epidemic routing, wherein every node is assumed to approach every other node in the system for a given time slot. The source node

passes the message on to a random receiver, which holds the message until it comes in contact with the destination node. Therefore, a message only makes a maximum of two hops, and overhead is reduced, at the cost of an increased delivery time.

Infostations and SWIM: In [14], Iacono et al propose a model where users can connect to the network in the vicinity of Infostations, which are distributed throughout the network area, and connected to each other. These Infostations provide very high data rates to users in their vicinity, but do not extend over a large area. Therefore nodes that may venture outside the coverage of the Infostation experience intermittent connectivity, and must always transmit only when they come in contact with the Infostation. In [33], Small et al propose a shared infostation model, where multiple Infostations may receive the user data, and serve as the destination node. Therefore, message propagation is similar to that in Epidemic routing. This approach reduces the time taken for messages to be delivered, but comes at the cost of increased network capacity usage.

Mobile Relay Protocol: In [26], Nain et al propose the Mobile Relay Protocol, which integrates message routing and storage in the network. In this protocol, if a node is unable to route directly to its destination, it issues a local broadcast to its immediate neighbors, which in turn check for a path to the destination. If they are unable to find one, the message is stored in a node's buffer. The protocol defines specifics on when messages are stored and removed from these buffers.

Spraying: In [35], Tchakountio et al propose a Spraying protocol in which message forwarding is restricted to the *last known direction* of the destination, which is kept track of by a location manager. In highly mobile networks, this technique exploits the possibility that the destination node is still in the vicinity of its last known location. The message is first unicast to a node close to the destination's original location, and then multicast (sprayed) from that point. It is possible that the destination may receive duplicate messages, and an end-to-end mechanism discards those duplicates.

3.2.2 Estimation-based approaches

In these protocols, nodes estimate the probability, for each outgoing link, of eventually reaching the destination, instead of blindly forwarding messages. Based on this estimation, nodes decide whether to store the packet and wait for a better chance, or decide to which nodes and when to forward.

Next-hop information based approaches: These approaches only estimate the likelihood of transferring a message to the next hop node.

- In [6], Davids et al extend epidemic routing to include probability of delivery to each message. When two nodes meet, they compare the list of messages that they carry. A node then removes bundles for which the other node has a greater delivery probability. Further, these values degrade over time.

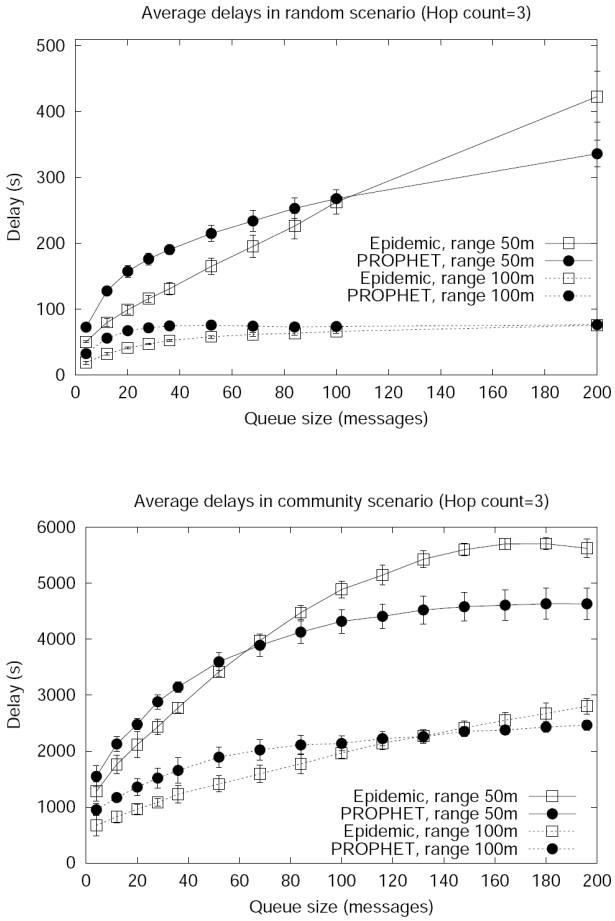

Figure 5: Performance comparison of PROPHET and Epidemic routing, reproduced from [23]

- In [23], Lindgren et all propose a probabilistic routing protocol called PROPHET. PROPHET first estimates a probabilistic metric called delivery predictability, *P(a, b)*, at every node *a*, for each known destination *b*. When two nodes meet, they exchange summary vectors (similar to lists in the above approach), and a delivery predictability vector. The delivery predictability is aged by a factor over time. Simulation results show that for the network considered, the improvement of packet delivery ratio under PROPHET over the epidemic routing can be up to 40 percent. This is shown in Figure 5.
- In [25], Musolesi et al introduce a Context-Aware Routing (CAR) protocol that integrates synchronous and asynchronous mechanisms for message delivery. The difference between these mechansims is the assumption that in synchronous delivery, when a packet arrives, a path to its destination exists, while in asynchronous delivery, no path exists at that time instant and the packet has to be stored for later delivery. If synchronous delivery is not possible, the protocol uses a time-series analysis technique - Kalman Filter - to predict the context information of the host with highest probability to deliver the packet to the destination. This protocol is particularly effective with limited buffer capacity.
- In [20], Juang et al experiment using wireless sensor nodes as collars attached to zebras to collect terrain information and study animal behavior. These sensors periodically report information when they come within range of a data collection object. The authors study two protocols - flooding and history-based routing - and find that flooding is more effective when buffer capacity is large, but consumes 8 times as much power as history-based routing.

End-to-end information based approaches: These protocols develop approaches based on observed end-to-end performance metrics of the network.

- Extending the work of [6], Burns et al propose the meets and visits (MV) protocol in [2]. This protocol uses the same exchange mechanism as in [6], but use a new method to estimate the likelihood of forwarding. MV learns the frequency of meetings between nodes and visits to certain regions, and use these frequencies to rank each bundle according to the likelihood of delivering a bundle through a specified path.
- In [34], Tan et al propose Shortest Expected Path Routing (SEPR). SEPR first estimates the link forwarding probability based on history data using a formula, and calculates the shortest expected path from it. When two nodes meet, they use this estimate to decide which node will forward the messages. Numerical results indicate that under SEPR, a 35 % improvement of delivery rate and 50 % reduction in resource cost can be achieved compared with epidemic routing [36] and routing in [6].

3.2.3 Model-based approaches

These protocols are based on the assumption that devices follow certain known patterns of mobility, thereby making it possible to estimate with greater accuracy which nodes are closer to the destination.

In [1], Becker et al present an approach called Model Based Routing (MBR), which uses world models of the mobile nodes for a better selection of relaying nodes and the determination of a receiver location without flooding the network. World models contain location information and user profiles indicating the motion pattern of users. These user profiles will also indicate the probability with which a relay moves towards the destination. However, no details are provided on how the authors plan to obtain the user profiles - they assume that receiver location is provided by a central service.

In [5], Chen et al model nodes moving along a highway. Although network partitions become significant in times of low traffic density, the fact that vehicles exhibit predictable behavior is used to relay messages in a store and forward fashion. Messages are propagated greedily each time step by hopping to the neighbor closest to the destination. Two kinds of transmission schemes are used, pessimistic forwarding and optimistic forwarding, which are distinguished by how long the messages are permitted to stay in intermediate nodes.

3.2.4 Node Movement Control based approaches

In the previous sections, the assumption made about the network is upon disconnection, the hosts passively wait for the network to reconnect. This leads to delays that may not be acceptable for some applications. The works discussed in this section propose techniques to reduce this delay by leveraging, and sometimes controlling node mobility.

- In [22], Li et al explore the possibility of varying node trajectories to facilitate communication in ad hoc networks. In contrast to letting the mobile host wait passively for reconnection, the mobile hosts actively modify their trajectories to minimize transmission delay of messages. The protocol is proposed as an application-layer one rather than a network layer one. Algorithms that minimize the trajectory modifications are developed under two different assumptions: the movements of all the nodes in the system are known, and the movements of the hosts in the system are not known. In the first case, a shortest path approach is used. In the second, hosts inform each other of their current position using a minimum spanning tree structure.
- In [8], Dolev et al propose Virtual Mobile Nodes (VMN), a distributed algorithm that runs on abstract nodes that move in a predictable manner. (VMN) travel through the network, collecting and delivering messages. In order to send a message, a real node examines its current location and calculates the current location of the VMN that is carrying out the service. The node then waits until the virtual node is nearby and transmits the message to the virtual node.
- In [38], Zhao et al propose a message ferrying (MF) ap-

proach to data delivery. This protocol utilizes a special set of mobile nodes called *message ferries* that move around the network coverage area and provide communication services for nodes in the network. Two schemes, Node-Initiated MF and Ferry-Initiated MF are detailed in this work. In the former scheme, ferries move in fixed path around the deployed areas, and nodes periodically move close to them to pass messages. In the latter scheme, when nodes want to send messages, they generate a service request, which the nearest ferry meets by moving towards the node. In [39], the authors extend the work by introducing multiple ferries to minimize average message delays.

- In [3], Chatzigiannakis et al present a snake protocol, where a snake-like sequence of carriers (called supports) or virtual nodes always remain pair-wise adjacent and move in a way determined by the snakeâĂŹs head. The head executes random walks over the area covered by the network, and the supports sweep the entire motion graph. They extend this work in [4], where each carrier performs a random walk sweeping the whole area covered by the network, which is the only difference between the two protocols. This protocol was shown to be more efficient (smaller message delays and memory requirements) and robust than the snake protocol.

- In [32], Shah et al propose a three-tier architecture called DataMules, that connects spare sensors at the cost of high latency. The top tier consists of access points which can be set at convenient locations. The middle tier consists of DataMules that are mobile nodes (whose mobility pattern is not known) and can communicate with sensors and access points. The bottom tier consists of sensors that are randomly distributed across a region. DataMules can pick up data from sensors when in close range, buffer it, and drop off the data to wired access points when in proximity. Numerical results establish relationships between buffer requirements at the sensors and DataMules and the number of sensors to be used.

We refer the reader to [37] for an excellent summary comparing the relative merits of each of the above mentioned protocols.

4. SECURITY

As the resources in a DTN are very limited, it is of great importance to restrict access to these networks, and ensure that unauthorized parties are not allowed to transmit on the network, and prevent authorized applications from accessing a higher class of service than they are entitled to. In this section, we observe some of the security considerations in LTP [16] and the bundle layer [15], and discuss research providing solutions for these issues.

4.1 LTP Security

Since LTP is a point-to-point protocol its security considerations are simpler than for the bundle protocol, and are well addressed in [17] through the security extensions. As LTP is a point-to-point protocol, most security considerations could be taken care of at another layer, either above LTP or in the link layer beneath. For this reason LTP does not define a confidentiality mechanism, but only data integrity mechanisms.

The LTP Authentication mechanism is an LTP segment extension comprising a ciphersuite identifier and optional key identifier that precede the segment's content, plus an authentication value (either a message authentication code or a digital signature) that follows the segment's content. The ciphersuite ID is used to indicate the length and format of the authentication value. This mechanism assures the authenticity and integrity of the segment.

The LTP cookie mechanism is an LTP segment extension that uses a randomly chosen numeric value that precedes the segment's content. By increasing the number of bytes in a segment that cannot be easily guessed by an inauthentic data source, and by requiring that segments lacking the correct values of these bytes be discarded, the cookie mechanism increases the difficulty of mounting a successful Denial of Service attack on an LTP engine.

In addition, the serial numbers of LTP checkpoints and reports are required to be randomly chosen integers. This randomness makes it even harder to launch a successful DoS attack.

4.2 Bundle Layer Security

Since the bundle layer protocol is an overlay network, Farrell et al surmise in [11] that it is more vulnerable to attacks, as every lower convergence layer's security issues apply to the bundle layer.

Due to the resource scarcity that characterizes DTNs, unauthorized access and use of DTN resources is a serious threat. If an unauthorized application were able to control some DTN infrastructure, perhaps by attacking a routing control protocol, and use the network resources, the extra resource consumption could effectively cripple the network. Another threat is that if DTN nodes could unwittingly be used to assist or amplify resource consuming behavior for example by not detecting unplanned replays or other misbehaviors, it would again be disasterous to the network. Another potential threat to the operation of the bundle layer is the injection of extra bundles into the network.

As with most protocols, Denial of Service attacks can be launched on the bundle protocol. DoS attacks would be even more effective against DTNs due to the longer latencies involved in several links. Implementations will need to be careful in the usage of cryptographic mechanisms, as each such mechanism would itself be vulnerable to DoS opportunities.

Apart from these threats, the usual threats to confidentiality and integrity of bundles also exist, for example, changing the intended destination or a bundleâĂŹs control fields.

In view of these threats, the DTN Research Group defined a set of goals in [15] for the security mechanisms in DTN implementations. These include:

- Promptly prevent unauthorized applications from having their data carried through or stored in the DTN.
- Prevent unauthorized applications from asserting control over the DTN infrastructure.
- Prevent otherwise authorized applications from sending bundles at a rate or class of service for which they lack permission.
- Promptly discard bundles that are damaged or improperly modified in transit.
- Promptly detect and de-authorize compromised entities.

In order to define security services for DTNs, a distinction is made between the sender of a bundle and the security-sender for an application of one of these services. Similarly, a distinction is made between the bundle recipient and the security-recipient (or security-destination) for a given application of a security service. The security-sender is the DTN node that applies the security service, and the security-recipient (or security-destination) is the DTN node that is the target for the security service - the node expected to decrypt or do integrity checking. The bundle security protocol allows for fairly flexible combinations of application of the confidentiality and integrity services.

In [29], Seth and Keshav propose a security architecture based on Hierarchical Identity Based Cryptography. In [21], Kate et al propose a DTN Security and Anonymity architecture, also based on Identity-based Cryptography to provide efficient security mechanisms. They also provide an analysis of the computation times typically presented by both solutions, and demonstrates that their solution outperforms that of [29]. Due to space considerations, we refrain from detailing these approaches, and refer the reader to these works.

5. A CASE STUDY: KIOSKNET

In this section, we examine an implementation of a DTN in the scenario of a rural environment. **KioskNet**, proposed by Seth et al in [31], presents a comprehensive solution that uses the transportation system from a city to a village as a *mechanical backhaul* to transfer data between a village and an internet gateway. The authors not only developed an economically viable and robust system by extending the DTN architecture, but also proceeded to implement a prototype of this system in a rural region in India, and demonstrated its potential applications.

The design goals of the system primarily aimed at minimizing the cost of the system, while simultaneously providing reliable service, in the face of intermittent connectivity. The system was also intended to allow user mobility and support to multiple types of devices. Data privacy was another consideration, to make applications like online rural banking and e-government services feasible. The system looked to leverage low cost technology, such as cheap storage, readily available wireless network cards and services, and recycled cell phones as GPRS modems to provide a reliable control plane over the ubiquitous cellular networks.

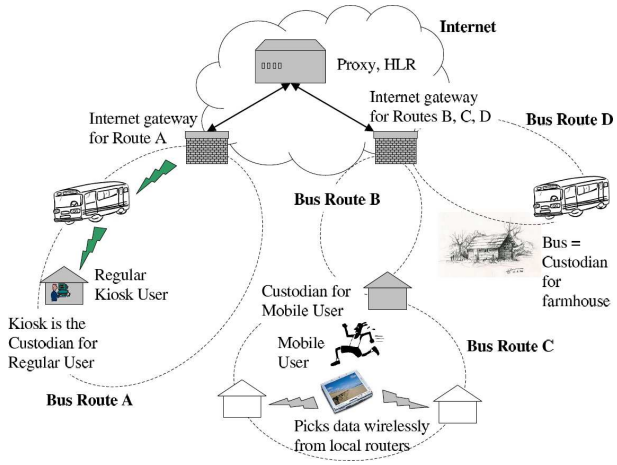

Figure 6: Sample KioskNet architecture, reproduced from [31]

Central to the architecture proposed by the authors is the kiosk, which consists of a *kiosk controller*, a server that provides network boot, a network file system, user management, and network connectivity by means of dialup, VSAT, or mechanical backhaul, or combinations of these. Kiosks serve primarily two types of users - regular users who connect through shared public access terminals in the form of recycled PCs, and mobile users who use the kiosk as a wireless hotspot providing store-and-forward access to the Internet. As mentioned earlier, internet connectivity is primarily provided through mechanical backhaul in the form of cars, buses, motorcycles, etc. that pass by the kiosk. These entities are required to possess a small, rechargable battery powered computer with about 40 GB of storage space and a wireless NIC, and communicate opportunistically with kiosk controllers. A sample architecture is illustrated in Figure 6.

Typical applications between the kiosk users and the Internet would include existing services like email, financial transactions, etc. Systems that provide such services are typically unable to deal well with delays and disconnections. The authors proposed the use of a disconnection-aware proxy to hide disconnection from legacy servers. The proxy is resident in the Internet and essentially has two halves. One half establishes disconnection-tolerant connection sessions with applications running on a recycled PC or on mobile userâĂŹs device, and the other half communicates with legacy servers.

The system includes mechanisms to provide secure, private communication through the use of hierarchical identity based cryptography, based on the authors earlier work [29]. Further, the system provides a simple API for application development that supports session persistence, intelligent use of multiple networks, and use of unmodified legacy servers. These features are provided by the use of Opportunistic Connection Management Protocol (OCMP) [30]. An OCMP client running on a mobile device can communicate opportunistically over multiple network interfaces to an Internet proxy. Legacy application protocols are hidden from the client through application-specific plugins that talk to legacy servers on behalf of the client.

The deployment of this system in Anandpuram, a village 20 km from the city of Vishakapatnam in India, is based on OCMP and the DTN-2 Reference Implementation (DRI). The system supports a number of standard and customized applications, most of which are written in a disconnection-tolerant fashion.

6. CONCLUSION

Delay Tolerant Networking has been a hot topic of interest in networking since the start of the century, and has sparked a significant amount of research in the area, particularly in an age where the ultimate goal is to provide ubiquitous connectivity, even in regions previously considered inaccessible. In this paper, we provide a comprehensive overview of Delay Tolerant Networking, and organize the paper in the form of a tutorial to provide readers unfamiliar with the field with a better overall understanding of the work done in the area, and bring them up to date with the state of the art to allow them to begin conducting research quickly.

DTN research is still relatively in its early stages, and there are still various open research problems in this field, and we have highlighted these wherever they have come up. We hope that our work will motivate more researchers to develop viable and efficient solutions in this field.

7. REFERENCES

[1] C. Becker and G. Schiele. New mechanisms for routing in ad hoc networks. In *4th Plenary Cabernet Wksp*, October 2001.

[2] B. Burns, O. Brock, and B. Levine. Mv routing and capacity building in disruption tolerant networks. In *INFOCOM '05*. IEEE, March 2005.

[3] I. Chatzigiannakis, S. Nikoletseas, and P. Spirakis. Analysis and experimental evaluation of an innovative and efficient routing protocol for ad-hoc mobile networks. *Lecture Notes in Computer Science*, 1982, 2001.

[4] I. Chatzigiannakis, S. Nikoletseas, and P. Spirakis. An experimental study of basic communication protocols in ad-hoc mobile networks. *Lecture Notes in Computer Science*, 2141, 2001.

[5] Z. Chen, H. Kung, and D. Vlah. Ad hoc relay wireless networks over moving vehicles on highways. In *MobiHoc '01*. ACM, 2001.

[6] A. Davids, A. H. Fagg, and B. N. Levine. Wearable computers as packet transport mechanisms in highly-partitioned ad hoc networks. In *International Symposium on Wearable Computing*, October 2001.

[7] M. Demmer, E. Brewer, K. Fall, S. Jain, M. Ho, and R. Patra. Implementing delay tolerant networking. Technical Report IRB-TR-04-020, Intel Research Berkeley, 2004.

[8] S. Dolev et al. Virtual mobile nodes for mobile ad hoc networks. In *18th International Symp. Distributed Comp. (DISC)*, 2004.

[9] K. Fall. A delay-tolerant network architecture for challenged internets. In *SIGCOMM 2003*. ACM, August 2003.

[10] S. Farrell and V. Cahill. LTP-T: A generic delay tolerant transport protocol. Technical Report TCD-CS-2005-69, Computer Science, Trinity College Dublin, 2005.

[11] S. Farrell and V. Cahill. Security considerations in space and delay tolerant networks. In *SMC-IT '06*. IEEE, 2007.

[12] M. Grossglauser and D. Tse. Mobility increases the capacity of ad hoc wireless networks. In *InfoCom '01*. IEEE, 2001.

[13] R. Handorean, C. Gill, and G. Roman. Accommodating transient connectivity in ad hoc and mobile settings. In *Pervasive '04*, April 2004.

[14] A. Iacono and C. Rose. Infostations: New perspectives on wireless data networks. Technical report, WINLAB, Rutgers University, 2000.

[15] IRTF. RFC 4838 - DTN Architecture, 2007. http://www.ietf.org/rfc/rfc4838.txt.

[16] IRTF. RFC 5325 - Licklider Transmission Protocol, 2008. http://www.ietf.org/rfc/rfc5325.txt.

[17] IRTF.R RFC 5327 - Licklider Transmission Protocol - Security Extensions, 2008. http://www.ietf.org/rfc/rfc5327.txt.

[18] IRTF. Delay Tolerant Networking Research Group, 2009. http://www.dtnrg.org.

[19] S. Jain, K. Fall, and R. Patra. Routing in delay tolerant networks. In *SIGCOMM '04*. ACM, 2004.

[20] P. Juang. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet. In *ASPLOS*, October 2002.

[21] A. Kate, G. Zaverucha, and U. Hengartnerl. Anonymity and security in delay tolerant networks. In *SecureComm '07*. IEEE, September 2007.

[22] Q. Li and D. Rus. Communication in disconnected ad hoc networks using message relay. *J. Parallel Distributed Computing*, (63), 2003.

[23] A. Lindgren, A. Doria, and O. SchelÃl'n. Probabilistic routing in intermittently connected networks. *Mobile Computing and Communication*, 7(3), July 2003.

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

[25] M. Musolesi, S. Hailes, and C. Mascolo. Adaptive routing for intermittently connected mobile ad hoc networks. In *Workshop on Wireless Mobile and Multimedia Networks (WoWMoM)*. IEEE, 2005.

[26] D. Nain, N. Petigara, and H. Balakrishnan. Integrated routing and storage for messaging applications in mobile ad hoc networks. In *WiOpt*, March 2003.

[27] E. Oliver and H. Falaki. Performance evaluation and analysis of delay tolerant networking. In *MobiEval '07*. ACM, June 2007.

[28] A. Petz and C. Julien. An adaptive middleware to support delay tolerant networking. In *ARM 2008*. ACM, December 2008.

[29] A. Seth and S. Keshav. Practical security for disconnected nodes. In *ICNP Workshop on Secure Network Protocols*. IEEE, 2005.

[30] A. Seth, S. Keshav, and S. Bhattachharaya. Opportunistic data transfer over heterogeneous wireless access networks. http://blizzard.cs.uwaterloo.ca/keshav /home/Papers/data/05/ocmp.pdf.

[31] A. Seth, D. Kroeker, M. Zaharia, S. Guo, and S. Keshav. Low-cost communication for rural internet kiosks using mechanical backhaul. In *MobiCom '06*. ACM, September 2006.

[32] R. Shah. âĂIJdata mules: Modeling a three-tier architecture for sparse sensor networks. In *SNPA Wksp*. IEEE, May 2003.

[33] T. Small and Z. Haas. The shared wireless infostation model âĂŤ a new ad hoc networking paradigm (or where there is a whale, there is a way). In *MobiHoc '03*, June 2003.

[34] K. Tan, Q. Zhang, and W. Zhu. Shortest path routing in partially connected ad hoc networks. In *Globecom '02*. IEEE, 2002.

[35] F. Tchakountio and R. Ramanathan. Tracking highly mobile endpoints. In *Workshop on Wireless Mobile and Multimedia Networks(WoWMoM)*. ACM, July 2001.

[36] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. Technical Report CS-200006, Department of Computer Science, Duke University, 2000.

[37] Z. Zhang. Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: Overview and challenges. *IEEE Communications Surveys and Tutorials*, 8(1), March 2006.

[38] W. Zhao, M. Ammar, and E. Zegura. A message ferrying approach for data delivery in sparse mobile ad hoc networks. In *5th ACM IntâĂŹl. Symp. Mobile Ad Hoc Net. and Comp.* ACM, 2004.

[39] W. Zhao, M. Ammar, and E. Zegura. Controlling the mobility of multiple data transport ferries in a delay-tolerant network. In *INFOCOM '05*. IEEE, 2005.