An Architecture to Implement the Bundle Layer Function of Delay Tolerant Networks

Debora Helena Job
Instituto Militar de Engenharia - IME
Praça General Tibúrcio 80 - 22290-270
Rio de Janeiro - RJ - Brazil
deborahelenajob@gmail.com

Mariana A. N. Silva
Instituto Militar de Engenharia - IME
Praça General Tibúrcio 80 - 22290-270
Rio de Janeiro - RJ - Brazil
malencastro@gmail.com

Wallace Anacleto Pinheiro
Instituto Militar de Engenharia - IME
Praça General Tibúrcio 80 - 22290-270
Rio de Janeiro - RJ – Brazil
awallace@ime.eb.br

Ronaldo Moreira Salles
Instituto Militar de Engenharia - IME
Praça General Tibúrcio 80 - 22290-270
Rio de Janeiro - RJ – Brazil
salles@ieee.org

***Abstract*— Some networks formed by mobile devices are set to operate under high mobility, low battery autonomy and little or no infrastructure, leading to a challenging scenario of frequent delays and disconnections. Such scenarios motivated the creation of the DTN (Delay-Tolerant Networking) model including a new layer to the TCP/IP architecture: the bundle layer. This work proposes adaptations on the architecture and standards of the electronic mail so as to behave as the bundle layer of the DTN model, providing the necessary functions to store, carry and forward messages from their point of origin to their final destination. The proposed model was tested and brought satisfactory results, providing low loss and high delivery rates of messages in the network. Using such e-mail features, it was possible to set up the environment smoothly, providing ease of configuration along with reasonable results.**

***Index Terms*— delay tolerant networks, bundle layer, electronic mail, architecture.**

I. INTRODUCTION

In recent years, the growth of communication and computing mobile devices production, such as cell phones, PDAs (Personal Digital Assistants), portable computers, notebook/netbook and others, are contributing to the appearance of new network environments. The networks formed by this type of devices may be subjected to some important constraints such as high mobility, low battery autonomy and little or no infrastructure. These factors contribute to the arising of challenging scenarios, where delays and disconnections cause inadequate functioning of the network.

Totally or partially disconnected environments that share a common inability to establish and preserve an end-to-end connection, aiming low packet loss and low latency in the network are known as DTN (Delay-Tolerant Networking) [1] or CHANTS (CHAllenged NeTworkS) [2].

Communication environments that present delays and disconnections, either for lack of resources or for the great mobility of devices, such as some types of sensor networks and MANETs (Mobile Ad Hoc Network), are proper scenarios for the operation of the DTN architecture [3]. This architecture can also be used in environments of rural, underwater, disaster, military and interplanetary communications.

In general, the DTN architecture provides a common method for interconnecting heterogeneous networks. It applies the paradigm of *store-carry-and-forward* to route messages in a journey from source to destination. In fact, the transferring service enjoys some similarity with the electronic mail service in data networks [4].

The problem observed in DTNs is that, because it is a relatively new subject, several issues related to the Bundle Layer (a.k.a Aggregation Layer) have not been adequately treated. This layer is one of the main contributions of DTNs, being able to perform the functions of storage, forwarding and custody transfer, among others. All these features have not been standardized and are still considered open questions. Current works on DTNs are implementing these features in the top layer, the application layer, making it difficult to reuse in other components of the DTN architecture.

Therefore, the goal of this paper is to employ the architecture and standards used by electronic mail, which have been tested extensively and are considered reliable and robust, to create a communication platform for DTNs with few adjustments. That is, to develop a model using the e-mail architecture and to apply it to the aggregation layer of DTNs.

This paper is divided into six sections. Section 2 examines the main concepts of delay and disruption tolerant networks – DTN, and details the main features of electronic mail architecture and as well as the similarities between the two architectures: DTN and e-mail. The related works are presented in section 3. Section 4 explains the proposed architecture and its implementation. Section 5 presents the experiments and the corresponding evaluation results. The conclusion of the paper is presented in section 6.

II. RELEVANT CONCEPTS

A. DTN

The Internet success is due to the smooth operation of TCP/IP protocols in environments with available network technology. Nowadays, there are many technologies such as satellite, optical fiber, 3G and 4G, and others that make communication possible. But there are environments in which the Internet presents itself limited either by shortage or lack of resources, or environments where it is difficult to establish and maintain an end-to-end communication session with low loss and latency. These environments are also known as challenging environments [1].

It is in these challenging environments that delays and disruption tolerant networks (DTN) appear as a viable solution for data communication [5, 6].

The DTNs' architecture was proposed by the research group DTNRG (DTN Research Group) [7] described in RFC 4838 [6]. The architecture has several characteristics, such as addressing scheme, the oracles of knowledge, aggregation layer etc., that are being evaluated by researchers.

The problem of delays and frequent disconnections observed in some particular scenarios is the key to the whole development of the DTN architecture. Delays in DTNs can take hours or even days, caused by both signal propagation and transmission of the message. Disconnections, on the other hand, can be originated by several different reasons, such as: high nodes mobility, awful conditions of communication channels, denial of service in order not to smudge the frequency in military operations, and economy of energy resources of the devices [8].

To solve the problems related to DTNs, it was proposed the use of message switching technology with persistent data storage. The message switching technique is one of the main paradigms for communications, where it is not necessary to establish a dedicated path between source and destination. Fig. 1 illustrates how a message is sent in standard network environments, where an end-to-end connection is established between source and destination, and also outlines the dynamics involving message switching hop-by-hop, by using intermediate nodes with persistent storage. The storage capacity is relevant for DTN performance once large amounts of messages may be stored in the node.

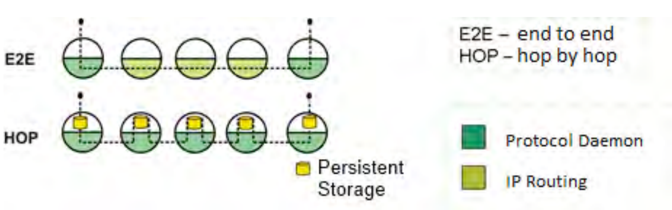

Fig. 1 - Forms of forwarding messages. Adapted from [1].

It is noteworthy that DTNs use opportunistic contacts to perform messages routing, noting, thus, that the end-to-end semantics, inherited from the reliability of TCP, becomes obsolete for this type of environment, once all the network routing occurs hop-by-hop.

Another approach about opportunistic contacts is referred in [9]. Social behaviors are taking into account in order to determine which nodes are to be reached as servants in a file sharing application. In this case, however, the authors do not use the bundle protocol because neither the node that generated the task nor servants know which node will be the destination of that task. Yet, in order to transmit messages among networks, the DTN paradigm was strongly considered.

Fig. 2 provides an overview of the layers model proposed for DTNs. It can be seen the behavior of messages or aggregates traversing each layer, where each intermediate node that has the responsibility of forwarding the message is named custodian [4]. The custodian node stores and then forwards messages to the next custodian, until it reaches the destination.

The aggregation layer is the main contribution of the DTN architecture [10] because it uses the concept of persistent storage, which helps to combat data loss with network interruptions, and also includes the hop-by-hop transfer with reliable data delivery. This layer also presents a series of diagnostics and management features that are implemented by the BP (Bundle Protocol), which specifications are documented in RFC 5050 [11]. It is important to note that these specifications are still being analyzed and are not standardized yet.

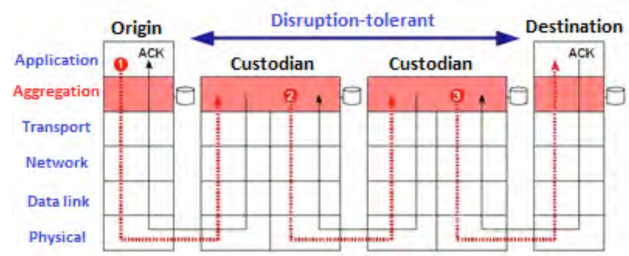

Fig. 2 - Layers Model proposed for DTNs.

To maintain interoperability, the aggregation layer uses a flexible naming scheme capable of encapsulating different addressing names and schemes, to minimize severe impacts that might occur among the different types of networks. As a protection, the aggregation layer allows the use of a security model optionally enabled [6].

In general, the routing protocols in DTNs allow storing prior knowledge of the network through several criteria that must be analyzed, for example, the last contact time, the frequency within which the node is present on the network in a window of time, how social [12] is one node in the network, etc. All these and other criteria will directly influence the routing to decide whether the node is good or not to carry the message, or find the best relay in the network.

The DTN architecture standardized certain terms, such as aggregates (Bundles) that are the messages that contain data in a DTN. Among them [1] [6] [11] [13]: custodian nodes, which are the nodes responsible for delivering an aggregate to another node, data mule which are the mobile nodes that allow network coverage and/or number of opportunities extension for communication (e.g., transportation, people or animals), endpoint that is a grouping of DTN nodes with similar abstraction to a multicast group (each point is identified by endpoint identifiers - ED - named using URIs), late binding, which is a technique that maps the path to its recipient's IP address as the message traverses the intermediate nodes of the network, service classes that provide specificity for the group (low priority, normal priority and express priority), types of contact to explain links structure between network nodes (classified as: persistent, on demand, scheduled, opportunists and predictable).

B. *Electronic Mail*

Electronic mail is made of two subsystems: one that enables reading and writing messages and another that performs their transfers. The first is called MUA (Mail User Agent), for example, Mutt, Pine, Outlook Express. The second, MTA (Mail Transfer Agent), which is the daemons of the system, i.e., processes that run in background and that are responsible for performing most of the work of the mail architecture, such as Postfix and Sendmail.

In general, the transfer and delivery of messages occur when a client using a MUA email creates the message and specifies the address of the recipient. This MUA forwards the message to its MTA, which determines whether or not it should receive the message. An MTA typically accepts messages for its own local users; to other’s systems, action known as forwarding; to users, systems, or networks that are allowed as messages relays. The MTA accepts and forwards the message to the MDA (Mail Delivery Agent), which sends it to a storage area when the message recipient is part of its domain; otherwise, the MTA passes the message to other intermediate MTAs until the message reaches its destination. If the MTA is not able to deliver the message or pass it on, it returns a message to the original message sender, or notifies the system administrator about it [14].

The SMTP protocol, using RFC 2821, specifies how the mail runs the transfer of messages on the network. This protocol is an ASCII text format, which creates a communication between client and server by establishing a TCP connection.

The proper functioning of electronic mail requires a permanent Internet connection. From the moment that the Internet access is done via modem, the connection is no longer end-to-end between the source and destination, and SMTP presented some problems. Besides, message boxes had to be developed in the MTA so that remote users could have access to their messages. This is how two other protocols were aggregated to the architecture of electronic mail: POP3 (Post Office Protocol version 3) and IMAP (Internet Message Access Protocol), that enable different features of the basic function of SMTP, which is the transfer of electronic messages.

C. *Similarities between: DTN and e-mail*

DTN architecture has many properties in common with e-mail [16].

DTNs work with message switching and message temporary storage, which is quite similar to the e-mail ability to attach some information that will be sent and stored on servers for later transfer to their destinations.

The architecture of e-mail is able to transfer messages between intermediate servers. This feature allows message transferring on the network by several hops. This is similar to the functionality of message passing hop-by-hop, which is necessary in DTNs.

Message transfer in DTNs has an optional service called custody transfer. This service makes the DTN node responsible for the delivery of the aggregate to its destination. This feature is present in the mail architecture, in the transfer of messages to a node set as a mail relay.

Regarding the assurance of message delivery, the reporting function in the e-mail architecture ensures the sender that its delivery is similar to the functioning of administrative records, that is used to provide information about the delivery of aggregates in the DTN.

In e-mail transferring, messages can be delivered in order of priorities. In general, there is no guarantee of the delivery time, but when a priority is set, the message becomes more urgent when compared to the others. Similarly, this functionality is used in DTNs, in the so-called classes of priorities, defining priorities for aggregates such as: low, normal and express.

Finally, there exists the option to fragment a large aggregate in DTN to ensure the delivery among contacts with known capacity of storage. In the e-mail architecture, MIME allows the fragmentation of a large message in several parts that are reassembled at the destination.

III. Related Works

A major consortium that develops projects and proposes standards related to the subject of this paper is the N4C (Networking for Communications Challenged Communities) [16]. This consortium aims to provide Internet access for people located in remote regions, i.e. in areas where communication usually has characteristics of long delays, high costs and high loss rates, which are caused by the lack of conventional network service. The projects of N4C consortium [17] [18] [19] focus their efforts on the application layer of the layer model described in FIG. 2. For the aggregation layer of this model, the projects use the Bundle Protocol [11]. Some example of projects developed by N4C are: Email/Not So Instant Messenger Applications, Web Caching Application services, Applications of Personal Digital Assistant – PDA – called Hiker's Applications, animal monitoring applications and transfer of meteorological data.

Goebbels presented Smart Caching [20], in which massive amount of data is be buffered in the user’s device whenever a good connection is available. Such approach makes the gaps of connection nearly transparent to users. Also, in periods of time when broadband connectivity is available, its usage was optimized, offering much more buffered data when possible. Similarly to e-mail functioning, it is possible to maintain data available (cached) while connectivity is low or inexistent.

Two other applications that use electronic mail in the application layer of the architecture are the DTN Brewer [25] and the Bytewalla project [21].

Among social projects that use DTN, DAKNet [22], Wizzy Digital Ccourier [23] and KiosKNet [24] should be emphasized.

It is important to consider that the related work described above attempts to provide communication between the points of origin and destination through the development of components for the application layer of the DTN architecture.

Fall's work [15] presents the fundamental concepts of the main applications for exchanging messages on the Internet. He stresses the importance of using DTNs to meet the growing need of permanent connection to the Internet, due to the increased use of mobile devices. Furthermore, there is a comparison between the message delivery systems and DTN architecture that, according to him, is similar to e-mail for having taken possession of several of its properties.

Fall's work [15] inspired the present research, for giving the initial subsidies that motivated the use of electronic mail, already established, directly in the aggregation layer of DTN's architecture, unlike the studies presented earlier that focus their efforts in the development of software artifacts for the application layer.

Therefore, it is extremely important to emphasize that the objective of our work is not to develop applications for e-mail to DTN as the projects presented before, but rather take advantage of the robustness and reliability in the present e-mail architecture and use it to compose the aggregation layer of the DTN architecture.

IV. PROPOSED ARCHITECTURE

The model designed in this work adapts the e-mail architecture to work under the *store-carry-and-forward* paradigm [3], necessary for environments with delays and frequent disconnections.

As mentioned earlier, for the e-mail architecture to behave properly in DTN environments, it is necessary to use the MTA as message relays. Fig.3 shows a simple example of the kind of environment considered in this work.

The environment is composed by mobile nodes subjected to opportunistic contacts [8], which take place through encounters between them, thus creating opportunities for communication. Also, we consider a network without a specific infrastructure with frequent disconnections. All network nodes are configured to act as message relays of each other, so that messages are stored and forwarded hop-by-hop. However, for the node to be a relay, it needs to be set for this function, which process will be described later on.

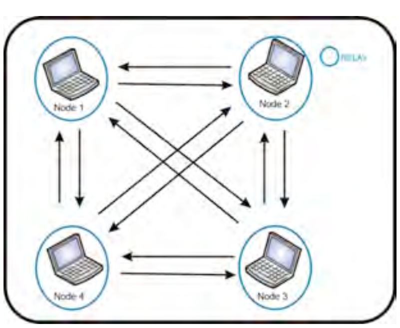

Fig. 3 – Example of network environment

According to [1], the aggregation layer is located between the application layer and transport layer of the network conventional layer model. The architecture proposed complies with this concept, and its location.

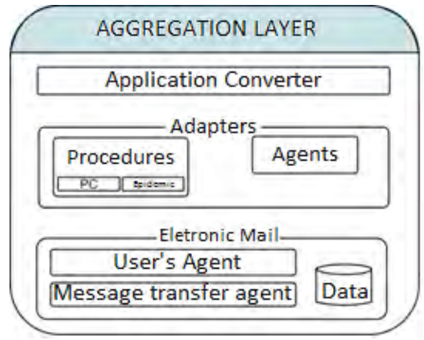

Fig. 4 - Magnified view of the sub layers of the proposed aggregation layer

Fig. 4 expands the sub layers belonging to the proposed aggregation layer, thus providing a better understanding of the idealized architecture. These sub layers are described in the following subsections.

A. E-Mail Sublayer Architecture

It consists of three modules: messages transfer agent - MTA, messages user agent - MUA and a base of data storage.

For the proposed architecture to work as the Aggregation Layer of DTNs, the settings of MTA to a local network were properly adjusted. Therefore, it is necessary that:

- The Message Transfer Agent - MTA is configured as relay messages to each network node;
- The User Agent - MUA is used to encapsulate the action of sending messages.

By adapting MTA to work as a relay to meet the needs of DTN environments, it was necessary to make changes to the MTA configuration files, to operate the lookup tables and also to manipulate the filters and message queues.

The lookup tables, or transport maps, are text files that map a key to search for a value that is usually indexed by lookup tables. The key must contain data, such as email addresses, domain or sub domains names. The values determine destinations and delivery methods, specified by the format (transport_type: next_hop), in which next hop indicates the node and port of message delivery, such as the IP address of a node [28]. These files are handled by the MTA.

The transport map of "transport" type was the option chosen for this work. It allows the e-mail sub layer to act as the relay, where the parameters for key, transport type and next hop on the transport map should be changed. Each network node must have a specified domain name. This is set as the key of the transport map. The type of transport receives the relay value and for the parameter next_hop, it gives the IP address of the node that will be its relay on the network.

Filters are commonly used for SMTP restriction: DNS block list, header expressions filtering, content filtering before and after queuing and delegation of access policies [26]. In the proposed architecture, we use the header_checks filter, which is a file that contains restrictions and actions that will be taken for each message. Its function is to compare the lines in the header of a message with a generated regular expression to prevent an action of being executed.

The message queue system is a method to queue and deliver incoming messages. Messages can be received by the mail server via SMTP or are locally sent and directly routed to the messages queue that are directories that indicate the current status of the message, during the process of queuing and delivery [27]. In the architecture, MTA makes the transfer of messages from incoming and hold queues to the active queue so the delivery process is started, and according to the destination addresses, the messages are delivered. But when a message is in a loop, this message is sent to the hold queue, which allows storing messages for a period of time until there is an opportunity to deliver them.

B. Adapters Sublayer

The Adapters sublayer provides the actions that e-mail sub layer does not do. These actions are to acquire information from the network, and from this information, change the research tables and message filters. This sub layer consists of agents and procedure modules.

The designed procedures perform actions in the MTA, MUA and also in the network. Their action in the network is directly linked to the awareness of the nodes that are reachable from other nodes. In MTA, they promote the update of the lookup tables according to the feedback received from the network, while in the MUA they act to compose the way messages are sent.

There are two procedures devised in this work called first contact (FC) and epidemic. The function of the FC is to work along with MTA to deliver the messages, through the first contact with the node, closest to it. The epidemic delivers messages to all available (directly reachable) nodes in the network, thus increasing the possibility of delivery to the destination.

The way the relay election is held is illustrated in Fig.5. Where, first (1) the FC procedure of the adapters sub layer performs a ping on the network, and waits for the response of all nodes that are at its reach. The responding node with the lowest RTT (Round Trip Time) is chosen as relay. After obtaining the network response, the FC procedure accesses the transport table (2a) and the header-checks table (2b). In the transport table, search keys are updated with the domain names of each network node, and the lookup values receive the type of transport as a relay, and the IP address of the elected node. In the header-checks, the regular expression that contains the chosen relay node address and the action of putting the messages in queue to avoid loops is written. The latter can occur if the message that has been recently passed by the node returns to it. After the tables have been changed, the FC calls the indexing program of the MTA, the postmap, (3a) (3b) in order to build the lookup tables. After the tables are indexed, the MTA forwards messages (4) to the elected relay, when the destination of the messages is out of reach.

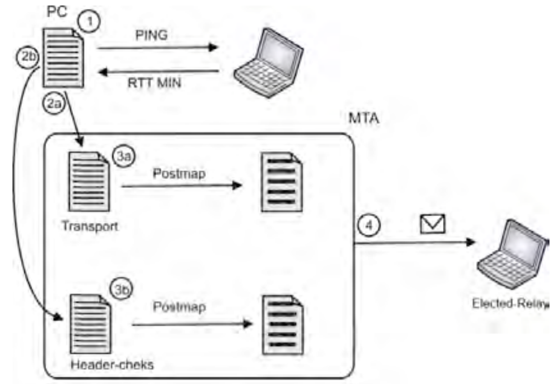

Fig. 5 - FC procedure operation of adapters sub layer

After the election of the relay, all nodes become message relays of each other. The effect of such procedure is to mimic the behavior of the custody transfers in DTNs. The node that is the relay of the source node receives the responsibility to forward the received message to another relay, when the destination node is out of reach.

C. Application Converter Sublayer

The application converter sublayer was designed to meet the requests of the Application Layer, such as sending a message and sending a file. Moreover, it can check the type of application requested and forward that request to the other sublayers of the aggregation layer.

This sublayer encapsulates the application layer requests and acts on the MUA of the mail sublayer, to perform the post.

D. Implementation of the Proposed Architecture

To make the machines work as nodes of a DTN, it was first necessary to select an MTA to be installed and adapted to the required functions. Among all MTAs available, Postfix SMTP was chosen. The reasons behind this choice are the fact that the software is free and open source, easy to use, flexible, modular, reliable, of high performance and widely used nowadays.

As said by [26], Postfix was developed in C language and runs on UNIX and Linux systems. It is a pure MTA, because it does not provide messages collection services, making such services the responsibility of POP and IMAP servers.

Postfix delivers messages to local users, but also relays messages from other systems, i.e., performs the function of MTA relay. However, by default, Postfix is not an open MTA relay to avoid UBE (Unsolicited Bulk Email) or SPAM [14].

For purposes of implementation, the architecture was designed to be used in a closed and controlled environment. Therefore, only the features of MTA on a local network will be addressed. Hence, all nodes that are part of the network are previously known and, therefore, issues of adding and removing nodes in the network will not be discussed.

The Postfix MTA has been installed, adjusted and its settings of transport maps and message filters were used to transfer messages to relay nodes and/or final destination. The adapters and application converter sublayers were implemented to be used together with de e-mail sublayer as presented in Fig. 4. Thus it was possible to accomplish a way of message relay node to node with persistent storage of data until the destination was reached.

V. Experiments

The need to verify the robustness of the proposed work led to tests conducted in an environment that represented, as close as possible, a real environment. In this context, experiments were performed on the infrastructure provided by the LABC2 project [28] at the Military Institute of Engineering (IME). This project works on an environment where potential network delays and disconnections can occur.

The architectures to be compared were installed on computers that could be used in tents, ambulances, hospitals and the Command and Control Center environments, described in [28]. However, to simulate different times of disconnections from the tents and ambulances, scripts were used to disconnect the network wireless interfaces. A specificity of the scenario is that at least once a day, the ambulances go from tents to hospitals and/or hospitals to hydration tents. Thus, there is a guarantee that the information is available with a maximum delay of twenty-four hours, as the ambulances can be used to make the role of relay nodes.

We used ten nodes in the network (8 notebooks, 1 netbook and 1 desktop). In tests, the notebook/netbook worked as mobile nodes and desktop represented the center of command and control (C2). A MANET has been assembled in a network where all messages were destined to the C2 node, being generated at each minute.

The experiments were performed using three different types of settings. First, the whole environment was set up using our proposed architecture (form now on called Mail_DTN), then the environment was set up with the bundle protocol (BP) provided by DTNRG [7], and finally, the environment was configured to use email without any adaptation (E-Mail), i.e., played the role of a mail server on a local network.

In order to simulate an environment of frequent disconnections between network nodes, we applied two different strategies: a deterministic one, where wireless interfaces were switched on and of at fixed time intervals; and a stochastic one, where such intervals were obtained from samples of probability distribution functions. These scenarios are presented in the following subsections.

A. *Fixed Time Intervals for Connection and Disconnection*

Intervals of 2, 3, 4 and 5 minutes were used to all nodes. Those periods were passed by value to the switching wireless interface procedure.

4 test-rounds were performed, lasting 1 hour each, for the 3 configurations, that is, one for the environment set up with the proposed architecture, a second one with BP and a last one with e-mail only, totalizing a 12-hour simulation for tests performed in fixed time intervals of connection and disconnection.

The graph on Fig. 5 presents the results obtained for average delivery and message losses in each configuration of the built environment (proposed architecture identified by Mail_DTN).

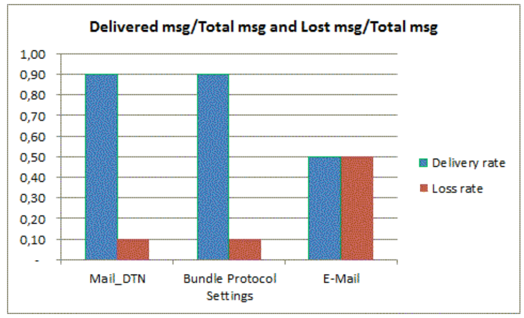

Fig. 5 - Test results of fixed time intervals of connection and disconnection.

The results indicate that in an environment with DTN features, the use of a solution that implements the *store-carry-and-forward* paradigm of is extremely important. The loss of information due to frequent disconnections in such environments is very expressive. This is evidenced by the results obtained for non-adapted e-mail architecture. For the tests with the proposed architecture and with BP, the results are pretty good, because all messages sent reached their destination, and those that did not, were stored in the nodes due to the end of simulation time.

B. *Uniformly Distributed Time Intervals for Connection and Disconnection*

To generate the time intervals of connection and disconnection using uniform probability distribution, the algorithm genunif.c was used, provided by [29].

It was decided to work with the time intervals of 3, 5 and 10 minutes. These times were passed to genunif.c as the maximum times of the interval. The algorithm in question also requires minimum time values. The settings take into consideration the following parameters: minimum time of 2 minutes, the file name to be generated, number of different seeds for each file and also the amount of variables that were generated. Table 1 presents these parameters. Therefore, for each node of the network it was generated three different files. Once 10 nodes were used in the test, 30 files containing time variables were generated. These were accessed by the wireless network interfaces switching procedure aiming to provide frequent disconnections in the network.

TABLE 1
INPUT AND OUTPUT PARAMETERS GENERATED BY GENUNIF.C

INPUT		OUTPUT
Key	Value	Generated variables
Output filename	uni_No1_T3	2,000008
PRNG seed	1	2,131538
Minimum Value	2	2,533535
Maximum Value	3	2,755605
Amount to generate	5	2,45865

Figures 6 and 7 present the results for delivery rate and loss rate, respectively.

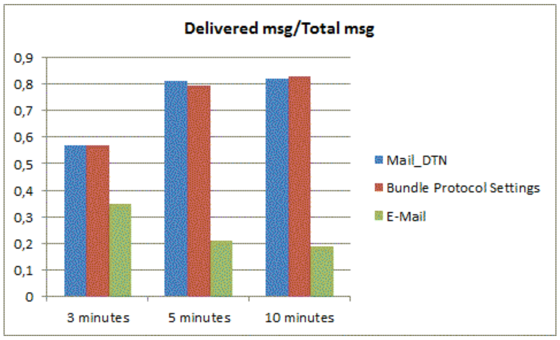

Fig. 6 - Delivery Rate

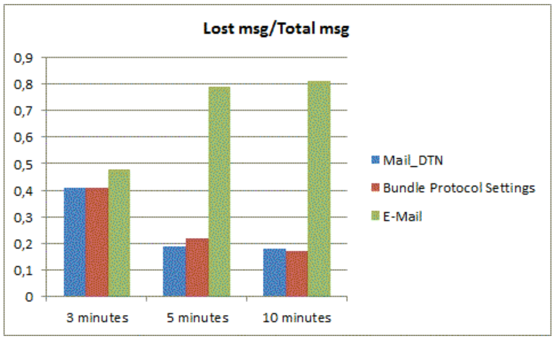

Fig.7 - Loss Rate

Comparing Mail-DTN and BP results, some small in performance can be noted. However, they are inconclusive and it is not evident which strategy is better regarding network performance only.

Thus, in general, it is clear that both Mail-DTN and BP are important strategies in challenging environments with frequent delays and disconnections. These strategies minimize losses and make networks more tolerant to these problems. However, due to the integration of trusted components of e-mail to its architecture, Mail-DTN shows some advantage.

C. *Exponentially Distributed Time Intervals for Connection and Disconnection*

Exponentially distributed samples for connection and disconnection times were generated using the genexp.c algorithm [29].

In this experiment, we decided to vary the expected time for connections (on) and disconnections (off) representing a wider range of disruption scenarios. For instance, in Fig. 8 scenario “[1-9]” represent a situation where the wireless interfaces are in average 1 minute *on* and 9 minutes *off* (a very disruptive scenario). On the other hand, scenario “[9-1]” is less disruptive since in average the interfaces remain 9 minutes *on* and only 1 minute *off*.

As shown in Fig. 8, the Mail_DTN results of delivery and loss rates are quite significant. Due to the large variation of connection and disconnection times of the network interfaces, it was possible to verify the proper functioning of the proposed architecture in situations of frequent disconnections. The worst performance was observed for scenario “[1-9]” and the best performance for scenario “[9-1]” as expected. Performance was improved almost linearly in between these two opposite cases. Thus, the Mail_DTN architecture provided consistent results.

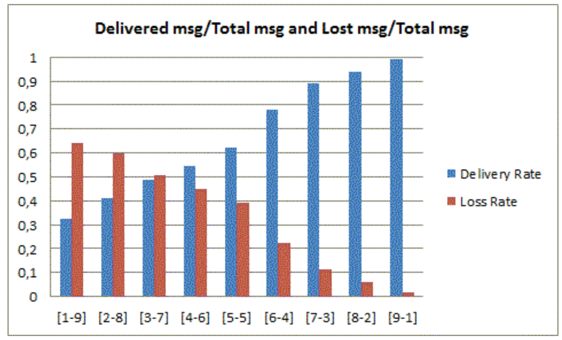

Fig. 8- Delivery and Loss Rates for the proposed Mail_DTN architecture

VI. Final Considerations

Electronic mail is a robust application with well-defined standards, extensively tested and very popular. It is a simple architecture, yet powerful, hence it supports various types of services, as the case of MIME types that allows images, videos and also voice files attachments.

The main objective of this study was to employ the architecture of electronic mail, which has similarities with the architecture supported by DTNRG, to create a platform for DTN communication. The architecture serves as an alternative implementation for the aggregation layer of DTNs. However, it is worth mentioning that e-mail stands out in relation to other alternatives by reasons of ease of configuration, reliability and robustness.

In this work, the architecture of electronic mail has been installed and properly adapted to perform the main function of the aggregation layer, storing and forwarding messages. Afterwards, experiments were performed on different emulated environments were the architecture was installed in netbooks, notebooks and desktop.

Through the experiments, it can be stated that the proposed architecture has handled, in an appropriate way, the storing and forwarding of messages in an environment with frequent delays and disconnections covered by DTN networks.

It is important to consider that architecture is an email application that has been adapted to operate as the aggregation layer. To allow e-mail application to operate on the proposed architecture, adjustments in the Application Converter sub layer have to be done, being the subject of future work.

References

[1] M. Demmer, E. Brewer, K. Fall, S. Jain, M. Ho and R. Patra. *Implementing Delay Tolerant Networking*. 2004. IRB-TR-04-020.

[2] L. Chen, C. Yu, T. Sun, Y. Chen and H. Chu. *A Hybrid Routing Approach for Opportunistic Networks*. In ACM SIGCOMM Workshop on Challenged Networks (Chants), ACM Press. p. 213-220, 2006.

[3] K. Obraczka. Lecture on Disruption-Tolerant Networks. Rio de Janeiro, UFRJ-COOPE. August 18, 2009. Course offered for students and teachers in COPPE-UFRJ

[4] K. Fall, W. Hong, S. Maddenr. Custody Transfer for Reliable Delivery in Delay Tolerant Networks. 2003.

[5] K. Fall, S. Farrell. Internet Research Task Force - IRTF. 2004. Available: http://www.irtf.org/dtnrg. Accessed in: July 20 2009.

[6] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, H. Weiss. Delay-Tolerant Networking Architecture. RFC4838.

Available: http://www.rfc-editor.org/rfc/rfc4838.txt. Accessed in: June 03 2010.

[7] K. Fall, S. Farrell. Delay-Tolerant Networking Research Group. DTNRG. Available: http://www.dtnrg.org. Accessed in: October 20 2009.

[8] C. T. Oliveira, D. M Taveira, R. B. Braga, Otto C. M. B. Duarte. Uma Proposta de Roteamento Probabilístico para Redes Tolerantes a Atrasos e Desconexões. 26th Brazilian Symposium on Computer Networks and Distributed Systems – SBRC. 2008.

[9] Palazzi, C. E. and Bujari, A. (2011), Social-aware delay tolerant networking for mobile-to-mobile file sharing. Int. J. Commun. Syst.. doi: 10.1002/dac.1324

[10] EM Daly, M. Haahr. The Challenges of Disconnected Delay-Tolerant MANETs. 2009. Ad Hoc Networks.

[11] K. Scott, S. Burleigh. Bundle Protocol Specification. RFC5050. 2007. Available: http://www.rfc-editor.org/rfc/rfc5050.txt. Accessed in: June 03 2010.

[12] G. Bigwood, T. Henderson. Social DTN Routing. International Conference On Emerging Networking Experiments And Technologies. 2008.

[13] K. Fall. The Delay-Tolerant Network Architecture for Challenged Internets. ACM SIGCOMM, ACM Press. 2003.

[14] KD Dent. Postfix: The Definitive Guide. 2003. 1. and d. O'Reily. 260 p. il. ISBN 0-595-00212-2.

[15] K. Fall. Messaging in Difficult Environments Delay Tolerant Networking as a Generalized Messaging Service. 2003.

[16] M. Udén. Networking for Communications Challenged Communities [online]. In 2008. N4C. Available: http://www.n4c.eu/Home.htm Accessed in: October 28 2008.

[17] T. Hyyryläinen and T. Kärkkäinen and C. Luo and V. And J. Jaspertas And J. Karvo Ott. Opportunistic Email Distribution and Access in Challenged Heterogeneous Environments. TKK Helsinki University of Technology, Networking Laboratory. 2007.

[18] A. Doria, M. Uden, DP Pandey. Providing Connectivity to the Saami Nomadic Community. 2nd Int. on Open Collaborative Design for Sustainable Innovation. 2002.

[19] M. Kindgren. DTNmail. 2007. Royal Institute of Technology (KTH), Laboratory at Kista. Sweden.

[20] Goebbels, S. (2010), Disruption tolerant networking by Smart Caching. Int. J. Commun. Syst., 23: 569–595. doi: 10.1002/dac.1088

[21] H. Ntareme, A. Doria, M. Zennaro, B. Pehrson. Bytewalla: Delay-Tolerant Networks on Android phones. 2010.The Royal Institute of Technology - CSD Lab

[22] A. Pentland, A. Fletcher, A. Hasson. DakNet: Rethinking Connectivity in Developing Nations. Computer. vol.37, no.1, p. 78-83. 2004.

[23] L. Wood, A. Rabagliati, D. Wood. Wizzy Digital Courier [online]. 2003. Available: http://wizzy.org.za/article/articlestatic/19/1/2/. Accessed in: October 06 2008.

[24] S. Keshav. The KioskNet Project [online]. 2007. Available: http://blizzard.cs.uwaterloo.ca/tetherless/index.php/KioskNet. Accessed in: October 06 2008.

[25] E. Brewer. Technology and Infrastructure for Emerging Regions [online]. 2007. Available: http://tier.cs.berkeley.edu. Accessed in: October 06 2008.

[26] M. Bäck, PB Koetter, R. Hildbrandt, A. McDonald, D. Rusenko C. Taylor. Servidor de Email Linux: guia de instalação, configuração e gerenciamento para pequenos escritórios. 2007. Pearson Prentice Hall.

[27] FP Brooks. The Postfix Home Page. [Online]. Available: http://www.postfix.org. Accessed in: January 07 2010.

[28] Salles, RM, Noya, RC, Bergman, U. Laboratory of Epidemiology Information Acquisition and Dissemination. LABC2. Military Institute of Engineering / FAPERJ, 2010. Available at: http://www.labc2.ime.eb.br/epidemio.htm.

[29] K. Christensen. Professor and Director of Undergraduate Program [online]. Department of Computer Science and Engineering. Available: http://www.csee.usf.edu/ ~ christen / tools / Accessed in: January 10 2010.