METhoD: a Framework for the Emulation of a Delay Tolerant Network Scenario for Media-Content Distribution in Under-Served Regions

Sandra Siby*, Adriano Galati†, Theodoros Bourchas†, Maria Olivares†, Thomas R. Gross*, and Stefan Mangold†
*Department of Computer Science, ETH Zurich, Switzerland
†Disney Research, Zurich, Switzerland

***Abstract*—Wireless communication remains to be the most efficient way of providing access to information to users in developing economies. We are interested in Delay-Tolerant Networks (DTNs) for distributing multimedia content to micro-entrepreneurs in under-served rural areas of South Africa. In our project, public commuter buses carrying WLAN-enabled devices (mobile infostations) that provide DTN connectivity are used to ferry data between urban and rural areas. Before the actual deployment in the field, rigorous lab experiments have been performed to study the performance of the proposed network design. Such an activity is time consuming and requires arduous preparation. Simulators, while convenient, do not provide the most realistic results. Given the complexity of DTN testing, we try to find a middle ground approach by building a mobility emulator testbed, which we use to emulate the network scenario. In this paper, we outline the design and implementation details of the network emulator. We then describe the experiments that were performed to study the impact of different network dynamics on content delivery, and their results. The results from the experiments are used to improve our current network design.**

Fig. 1: Buses downloading content from the bus station in the city of Pretoria. Concept art © Disney.

I. Introduction

The Global ICT statistics of 2014 by the International Telecommunications Union indicate higher penetration rates for mobile cellular and broadband subscriptions in developing countries as compared to fixed, wired, subscriptions [1]. Access to information is essential for economic growth and development in developing regions. The lower costs associated with wireless networks make them an attractive means of providing access to information for developing economies. However, there are challenges that arise when deploying connectivity solutions in these areas; the technologies are usually built keeping the developed world in mind. They are based on the assumption that there are reliable end-to-end paths, steady connections, and continuously available power sources, which is not always the case in rural areas of developing countries [2]. Delay Tolerant Networking (DTN) is a network architecture aimed at providing communication in situations where end-to-end connectivity is not possible. To ease the development process of DTN applications, the DTN Research Group (DTNRG) has defined an experimental network protocol for challenged networks known as bundle. A bundle is a protocol data unit of the DTN bundle protocol. Bundles are transported in a store-carry-forward manner by the nodes in the network. The layer that implements the bundle protocol operations acts as an overlay network over existing transport layers. The bundle protocol describes all the entities and operations of the bundle layer such as the description of the bundle, routing rules, processing, and security issues. The protocol specification is described in detail in RFC5050 [3]. DTNs may be a low cost alternative to traditional wireless networks. Our MOSAIC 2B project [4] aims to unleash opportunities for micro-entrepreneurs in rural areas of South Africa by providing them with entertainment and educational media content. DTN is used as a method for delivery of this content. MOSAIC 2B plans to use DTN to take content from urban areas and distribute it to recipients in the rural areas. However, before the actual deployment of the project, the entire network must be validated, and the software and hardware components must be tested to ensure that they work as expected. Such an activity is time consuming and requires an arduous work of preparation. Simulators, while convenient, do not provide the most realistic results. Given the complexity of DTN testing, we try to find a middle ground approach by building METhoD, a Mobility Emulator Testbed for DTNs, which we use to emulate the MOSAIC 2B scenario. In this paper, which follows results reported earlier [5], [6], we present the METhoD framework for the emulation of delay tolerant networks and show network performance in the MOSAIC 2B settings before the actual implementation in South Africa. An overview of the MOSAIC 2B project is provided in Section II. The METhoD testbed is presented in Section III and evaluated in Section IV. Section V describes an emulation of the MOSAIC 2B project and presents results. A brief summary of related work is provided in Section VI. Finally, Section VII concludes the paper.

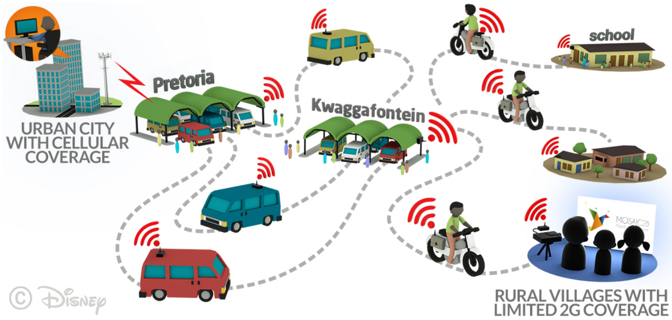

Fig. 2: The MOSAIC 2B scenario: A DTN network between the bus station in the city of Pretoria and rural bus station in Kwaggafontein.

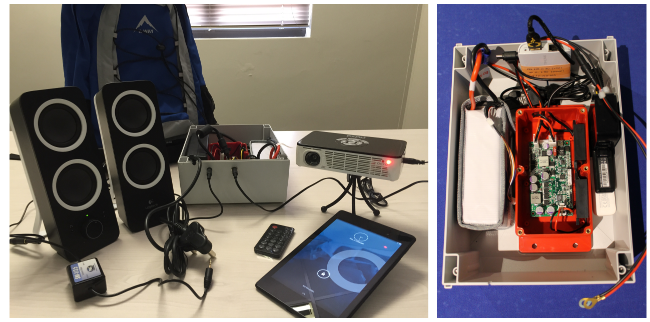

Fig. 3: Cinema-in-a-backpack system (left) and mobile infostation (right).

II. MOSAIC 2B Overview

MOSAIC 2B is a research project aiming to provide business opportunities for micro-entrepreneurs living in rural South Africa by delivering multimedia content to them in a low cost manner. Since cellular data access is usually unavailable or expensive in rural areas, content delivery will be performed using DTN. Figure 2 gives an overview of the project. Content is delivered from the city of Pretoria to Kwaggafontein, a small town located approximately 100 km from Pretoria which serves several rural areas. Buses usually travel between the two locations on predetermined paths, for approximately three hours in the morning and three hours in the evenings. The buses travelling between these two locations act as carriers of data. This can be achieved by placing low cost WLAN-enabled devices, named *infostations*, in the buses and bus stops at both terminal locations (see Figure 1). In the rest of the paper, we refer to the infostations placed in the stops as fixed infostations and the infostations placed in the buses as mobile infostations. Initially, content is stored in a fixed infostation located in Pretoria. When mobile infostations come in contact with the fixed infostations, i.e., when buses arrive at the bus stop, data is sent epidemically [7] to the mobile infostations via DTN (see Figure 1). The mobile infostations act as data mules and transport the content between the infostation in Pretoria and another fixed infostation in Kwaggafontein. Micro-entrepreneurs in the areas near Kwaggafontein are equipped with cinema-in-a-backpack systems that allow them to obtain content from the fixed infostation in Kwaggafontein and screen it.

The infostations (fixed and mobile) are wireless routers equipped with a USB hub, battery supply, external memory, GPS receiver and 3G dongle (see Figure 3). We use the TP-Link TL-MR3040 Version 2.0 as the router for the infostations. Since the router's memory is not sufficient for our purposes, we connect a USB hub to the router and use it to accommodate external memory storage. In addition to the external memory, a GPS receiver and 3G dongle are attached to the hub. The GPS receiver is used to obtain GPS coordinates of the infostation. The 3G dongle is used whenever cellular network is available to send information such as mobility traces and system status back to the central server in Pretoria. This setup provides some monitoring of the system and enables us to detect failures. The fixed infostations are connected to the power system of the bus stops, while the mobile ones can be eventually powered up by the battery of the vehicles.

The cinema-in-a-backpack carried by the entrepreneurs consists of the following components: a tablet, a projector, speakers and a battery. The tablet allows the entrepreneur to obtain the content from the fixed infostation at Kwaggafontein. The projector can be connected to the tablet via HDMI/VGA. A battery is provided for use when power supply is not available. The components of the cinema-in-a-backpack are shown in Figure 3.

III. METhoD Framework

There are several methods of testing DTN setups before deployment. Real-life testbeds provide the most accurate results. However, creating and running an actual testbed can be a time consuming and costly affair, requiring constant supervision. In addition to these drawbacks, testbeds involving mobility usually require large areas for producing node disconnections. A popular alternative is simulation. Most DTN applications are tested through simulators. Simulation can be easily and conveniently performed in the laboratory. However, most simulators make use of simple models and methods and do not capture various aspects of real life testing. A viable middle ground is an emulation approach. Rather than simulating the mobile nodes, we can use the actual wireless devices and replicate their movement in a testbed. This approach presents an improvement over simulation since we use the same hardware and software components that we intend to use in the real-world deployment. The design of the METhoD architecture is driven by the following goals: mobility emulation, DTN layer communication, and node cluster isolation. The testbed detects when mobile nodes are within range of each other. When nodes are in contact, they can connect and exchange data at DTN level. Besides, the testbed supports separate clusters of connected nodes. This property allows groups of nodes communicating with each other in an intra-cluster fashion at the same time. Keeping these requirements in mind, a centralized controller is designed to coordinate the location and connectivity of all the nodes in the testbed (see Figure 4). Every node has two connection interfaces: one interface is used to communicate with the controller, while the other one is used for communication with the other nodes in the testbed. GPS traces, representing node mobility information, are fed as input into the testbed. It can take GPS traces from existing datasets as input, provided that they are processed into the proper format that is accepted by the testbed. However,

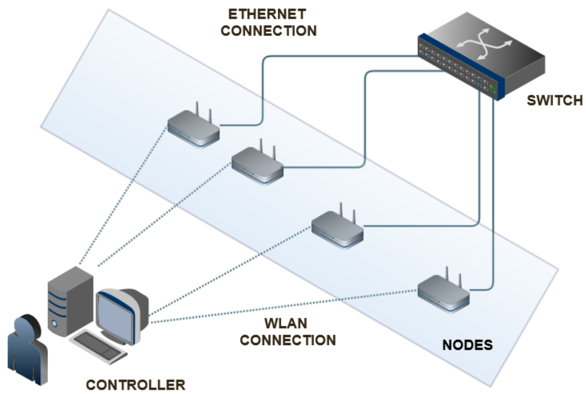

Fig. 4: METhoD testbed overview.

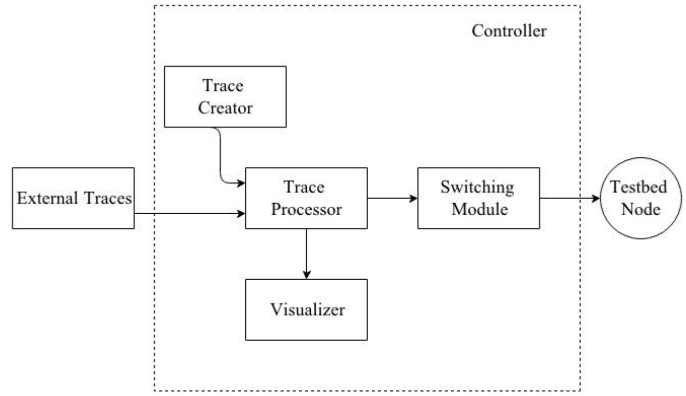

Fig. 5: The METhoD emulator framework.

for the emulation of the MOSAIC 2B scenario, GPS traces are generated to represent vehicles moving between Pretoria and Kwaggafontein. Such traces are used by the controller to calculate the time at which two nodes come in contact, so as to simulate their mobility. Connections and disconnections can be set up by changing the firewall rules of the devices on the fly. The mobile nodes are driven by the controller through the WLAN interface, while the data transfers among them take place via Ethernet cables connected to a 48-port switch. Figure 4 shows the overview of the METhoD testbed. The aforementioned configuration of interfaces is done to study the effects of node connectivity without taking into account the nature of the wireless channel. However, METhoD works even if the interfaces are switched. The controller is connected to the same ad-hoc network as the mobile nodes.

As shown in Figure 5, METhoD consists of four main components: the mobility trace generator, the mobility trace processor, the switching module and the visualizer. The trace generator is used to generate mobility traces. The trace processor takes the traces as input and produces connectivity traces. Such information is used by the switching module to run the actual test. The visualizer is a software tool to display the movement of the nodes. The functionality and implementation of the components are described in the following sections.

A. Trace Generator

METhoD provides a module to generate mobility traces of nodes. Mobility can be simulated using GPS traces obtained by drawing paths on an OSM map. The mobility trace creator can generate GPS traces for points placed along a path on a map. In addition to this, it can generate traces for vehicles of different speeds and start times, which gives us the flexibility to emulate different mobility scenarios.

B. Mobility Trace Processor

This module reads individual mobility traces and output connection information of each node. It achieves this by first extracting common timestamps between two nodes. Two nodes are never in contact if they have no matching timestamps. Then, the module checks during which timestamps the nodes are within a certain radio range of each other. It collates this information into time intervals of connections and disconnections. Connectivity traces are also generated in JSON format, which contains node ID, location and connection information for the visualizer.

C. Switching Module

The switching module performs several steps before the actual execution starts. First, it sets firewall rules on all the nodes. Then, it assigns the connectivity traces from the trace processor to all of them. Finally, the switching module reads the connectivity traces and spawns a new thread to handle it. The thread checks the connection status, if the status specifies that two nodes are to be connected, it creates a firewall accept rule to open a connection. If a disconnection is specified, it removes the existing accept rule for the relevant node. Once all of the nodes are ready, the emulation can start.

D. Visualizer

To better visualize the movement of nodes, a simple visualizer is provided. The JSON file containing node IDs, location and connection information, generated by the trace processor, is submitted to the visualizer. The visualizer plots the nodes on an OSM map. The Figure 6 shows two screenshots while reproducing the MOSAIC 2B scenario.

IV. Validation

Before using the testbed to perform experiments for MOSAIC 2B, we validated METhoD with the UMass dataset [8]. This dataset has been selected because, as for the MOSAIC 2B, it also involves mobility of buses. The dataset consists of connectivity traces of buses travelling in the University of Massachusetts, Amherst campus over a period of three years (2005-2007). The data was collected during the spring and fall months. Each file in the dataset represents connectivity of the buses during a single day. Mobility traces of fifteen buses travelling in the campus have been selected. The dataset has connectivity information over a period of 17.5 hours. The contact times of all of the nodes have been extracted by monitoring the pinging of each device with all the others. Although the experiment runs for the entire 17.5 hours and everything worked as expected, we noticed that sometimes nodes do not ping each other. In addition, some devices reboot

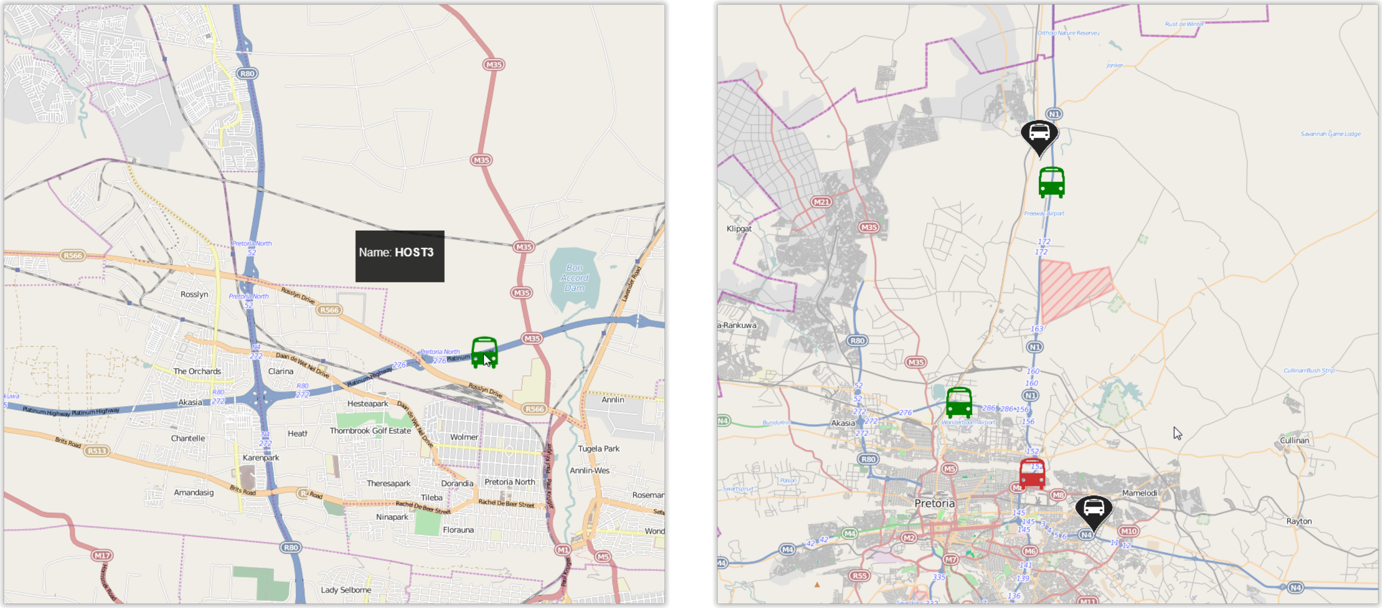

Fig. 6: Snapshots of the MOSAIC 2B scenario in the Visualizer of the METhoD emulator.

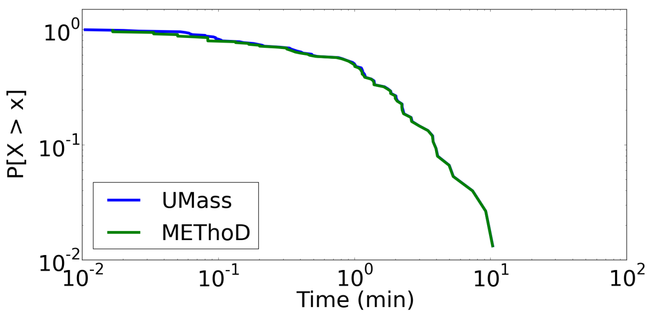

Fig. 7: CCDF of contact durations obtained from the UMass dataset and the METhoD emulator.

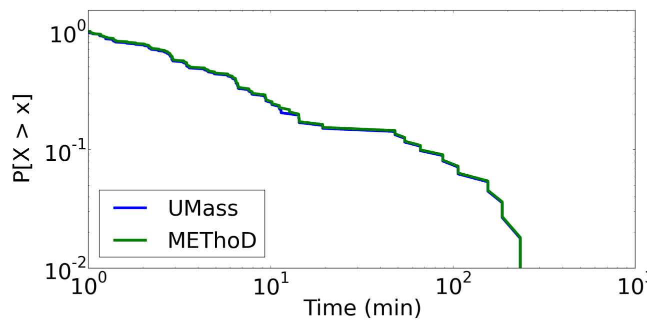

Fig. 8: CCDF of inter-contact durations obtained from the UMass dataset and the METhoD emulator.

occasionally. We notice that such nodes are the ones that have the highest number of connections, that is, they cannot handle many processes running on them. We confirm this conclusion by rerunning the entire experiment without launching the DTN services to reduce the number of processes, and the experiment successfully completes multiple times without any issues.

These results indicate that the number of processes running on a device needs to be taken into account during the deployment in South Africa. Figures 7 and 8 show the contact duration and the inter-contact time [9]–[12] based on ping activity for fifteen nodes. The contact duration between two nodes is the time during which they are in contact with each other. The inter-contact time of two nodes is the time between two consecutive contact durations of the same nodes. Figure 7 shows the CCDF of the contact time distribution generated by METhoD against the actual UMass distribution. Similarly, Figure 8 shows the CCDF of the inter-contact times of the METhoD against the UMass. In both plots, the distributions overlap with the real-world traces of the UMass dataset. The METhoD results are very good approximations of the UMass distributions, indicating that the connection setup occurs almost immediately, with negligible latency. The latency overhead, namely the excess time taken by METhoD to change the firewall rules with respect to the real traces, is 0.5 %.

V. MOSAIC 2B Emulation

Experiments to study the performance of the MOSAIC 2B network were presented in [5] and [6]. However, those experiments had shortcomings; outdoor experiments were time consuming and required constant supervision, while the indoor experiments involved stationary nodes and were not true representations of the MOSAIC 2B scenario. To perform an in-depth analysis of the system we built METhoD, a testbed that allows us to replicate the target scenario. We use the METhoD testbed to reproduce an environment similar to the one in MOSAIC 2B and run experiments (see Figure 9). In this way, we analyse the network performance in such settings and eventually identify and fix issues before the real deployment.

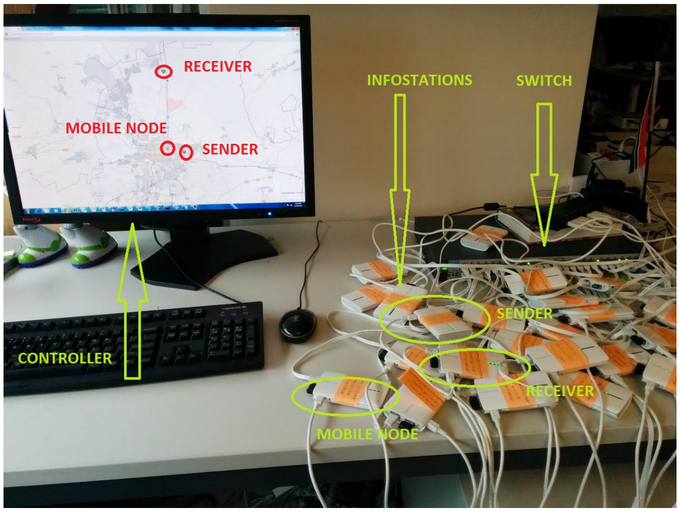

Fig. 9: Testing real devices in the testbed while tracking them on the visualizer map.

A. Experimental Setting

Since datasets of mobility traces of buses between Pretoria and Kwaggafontein are not available, we create our own traces using the testbed trace creator module provided by the METhoD framework. As an initial MOSAIC 2B scenario, we assume that buses wait for fifteen minutes at the Pretoria stop, travel towards the Kwaggafontein stop and wait there for fifteen minutes. Buses have a constant speed of 60 km/h. The route between Pretoria and Kwaggafontein is the one followed by the buses during the actual South African experiments. Mobility traces for five buses carrying the mobile infostations between two fixed infostations at the end points are generated. The time between two consecutive buses leaving an end point was one hour. The connection range between two infostations is derived using the Haversine formula [13]–[15]), which results to be about thirty meters. Such a distance guarantee a stable connection in real-world scenarios, where obstacles and the nature of the surrounding environment affect the communication link. The infostations are configured with OpenWrt, bleeding edge version (Barrier Breaker), a Linux-based operating system for embedded devices. They are also configured with IBR-DTN [16], [17], a C++ implementation of the bundle protocol targeted at embedded systems running OpenWrt. Since we want to observe data transfer at the DTN layer, we set up METhoD to turn on the IBR-DTN daemon on all the nodes. Initially, all of the nodes are set up with Epidemic routing [7]. To make the scenario more realistic, the link speed of the Ethernet interface has been changed from 100 Mb/s to 10 Mb/s using `ethtool`.

B. Emulation with a single movie

Initially, the source in Pretoria sends 1 GB (one gigabyte) data as a single bundle to the Kwaggafontein bus station. The MOSAIC 2B scenario considers transfer of large amount of data, that is, movies and multimedia content. The content is therefore forwarded to the mobile infostations mounted on the buses and carried to Kwaggafontein where it is forwarded to the final recipient. However, the contact time between the sender and the mobile nodes is not enough to transfer 1 GB bundles. Hence, reactive fragmentation occurs and the mobile nodes get part of the data. Notice that only 1 GB data file is considered for transmission, and all of the mobile nodes receive about 800 MB of it.

Figure 10 shows the transmission activity at the receiver during connection with each mobile node. Based on our previous results [5], which show proactive fragmentation expressing higher delivery times, the data file is provided as a whole bundle.

Since our previous results [5] show that proactive fragmentation produces higher delivery times, the first mobile node transfers the entire fragment to the receiver. Meanwhile, the second mobile node obtains a fragment and reaches the receiver. At this point, the data bundle stored at the receiver is forwarded back to the mobile node. This happens because both of the nodes store fragments of the original bundles identified by different signatures. The receiver assumes that the mobile node does not have its fragment, received by the first carrier, and forward it to the second one. Such a two-way data exchange behavior is seen in all the following meetings between the receiver and the mobile nodes. Notice that in the last two trips both nodes try to send data to each other, and ultimately, only a few hundred kilobytes are sent while in contact, which cannot be clearly seen in the plot. From this initial experiment, we observe that epidemic routing cannot be set on the receiver. One option would be to set up such a fixed node with 1-hop routing that allows transfer of data bundles only to the intended recipient.

Figure 11 shows the bundle activity at the receiver, which is set up with 1-hop routing, during connection with each mobile node. It shows that fifteen minutes is not enough time for the mobile nodes to obtain the entire 1 GB bundle from the sender. Hence, reactive fragmentation occurs and the fixed infostation in Kwaggafontein gets several fragments of the movie. Each mobile node transfers roughly the same initial part of the entire bundle. If the contact time is not sufficient to transfer the entire bundle, the DTN node may fragment a bundle cooperatively when a bundle is only partially transferred. A reactive fragmentation process occurs after an attempted transmission has taken place. A DTN node may also divide application data into multiple smaller blocks and transmit each block as an independent bundle. This approach is used primarily when contact volumes are known in advance. In the MOSAIC 2B scenario this is exactly the case, where buses follow a predefined timetable. In this first experiment, having no proactive fragmentation raises a new issue. Since the contact time is not sufficient to transfer the entire bundle, the receiver gets multiple overlapping large fragments. Eventually, it will receive the remaining parts when the same buses will be back in Kwaggafontein. Despite longer delays, proactive fragmentation can help to decrease the overlapping of data. Since mobile nodes carry fragments of the initial 1 GB data bundle, the receiver cannot distinguish between them so as to identify overlapping parts. DTN nodes must receive the entire bundles to be able to identify them. Enabling proactive fragmentation and splitting up the data file in smaller bundles, such that the forwarding time of a bundle is shorter than the contact duration between sender and receiver, reduce retransmission of data. In this case, disconnections do not occur while

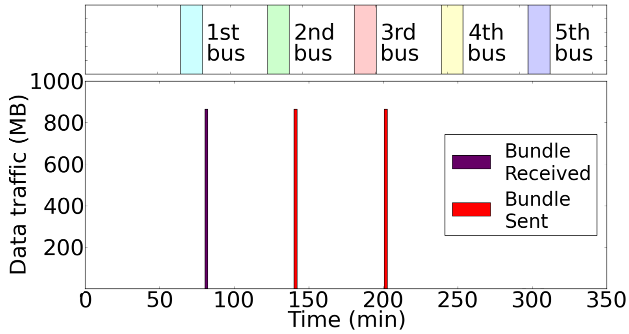

Fig. 10: Data traffic at the receiver. 1 GB movie transfer with no proactive fragmentation and receiver with Epidemic routing.

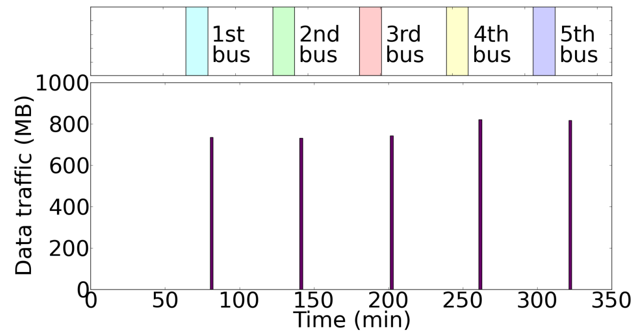

Fig. 11: Data traffic at the receiver. 1 GB movie transfer with no proactive fragmentation and receiver with 1-hop routing.

transferring the first bundle. Namely, at least the first bundle is not fragmented. Thus, future nodes carrying the same bundles stored by the receiver are not forwarded again. For this effect, we enable proactive fragmentation and split the 1 GB movie to be sent to Kwaggafontein in smaller bundles of 100 MB each. Figure 12 shows the bundle activity at the receiver when proactive fragmentation is enabled. Unlike the distribution in Figure 11, from the second contact onwards, bundles are not sent from the beginning. The proactive fragmentation enables the mobile node to send bundles not yet at the receiver. All of the nodes after the first one transmit only a small amount of data. This is because all of the nodes are at the sender for the same amount of time and receive the bundles in the same order. All the mobile nodes take approximately eight bundles. This means that when the nodes after the first one come to the receiver, they transmit only a fragment of a bundle. This is because each fragment of bundle is created from a reactive fragmentation, and it is not recognized as one of those bundles generated by the sender. Therefore, each mobile node produce a fragmented bundle if the contact time is not sufficient to transfer all the content. Thus, the efficiency we gain from using proactive fragmentation does not really help us since new bundles are never forwarded.

To ensure that the sender at the Pretoria bus station sends different sets of bundles to different mobile nodes, we make use of a scheduler to decide which bundles are to be sent at every contact. Such a scheduler provides two simple scheduling methods, random and FIFO. Figure 13 shows the effect of a random distribution of bundles by the sender, at the receiver side. It has a better performance with respect to the results in Figure 11 and Figure 12; all of the bundles get delivered over the five contacts. However, since the process of picking bundles is random, there is a chance that the same bundles could be picked during a contact, leading to a redundancy. In the FIFO distribution the bundles are sorted and sequentially forwarded to the carriers. The process is repeated when it reaches the end of the list of bundles. Figure 14 shows the effect of a FIFO distribution of bundles by the sender, at the receiver side. All the bundles are delivered within three contacts. In this case, we can better control which bundles are sent at which contact. Out of these first experiment, to get some quantitative results on which method performs better, we experiment with multiple movies in the following section.

C. *Emulation with multiple movies*

Since the MOSAIC 2B scenario involves sending multiple movies to recipients, we investigate the behavior of the system when more than one movie is sent. For that, three 1 GB movies are to be sent to Kwaggafontein. The devices are set up with proactive fragmentation of 100 MB size. After the bundles were created and stored on the sender, the test is kicked off. We first run the experiment without any scheduling at the sender side. The bundle activity at the receiver is shown in Figure 15 (left). After the first contact with the mobile infostations, only a fragment of the same bundle of the same movie is sent to the receiver at each contact. This means that irrespective of how many mobile nodes are there, they all receive the same bundles. Hence, unless the mobile node returns to the sender a second time, new bundles are not received at the receiver.

The same experiment is also repeated with random and FIFO bundle distribution. As shown from the results of the random distribution of bundles in Figure 16 (left) with respect to Figure 15 (left), a simple random distribution strategy can increase the efficiency. The FIFO distribution is the most efficient, as shown in Figure 17 (left). It leads to the delivery of the largest amount of data. This is because there is greater control over which bundles are chosen by the sender to transmit to the mobile node at each contact.

The delivery ratios of the three experiments are also presented next to each plot in Figure 15, 16, and 17, respectively. None of the three movies are delivered in one journey when there is no scheduling in place because all nodes get the same bundles. The random distribution has a much better performance and manages to deliver approximately 80% of all the movies. Finally, the FIFO distribution has the best performance; it almost succeeds to deliver all of the three movies. More precisely, it delivered two movies, and 90% of the third one.

VI. Related Work

Previous reports on the MOSAIC 2B project mainly consists of initial experiments to determine the performance of the system [5] and [6] and describe the effect of varying bundle sizes on delivery time under different network topologies. These reports help assess the system, but the experiments were time consuming. The testbed we propose aims to facilitate the

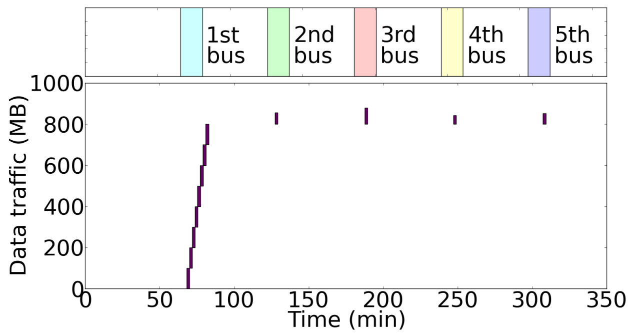

Fig. 12: Data traffic at the receiver. 1 GB movie transfer with 100 MB bundle size.

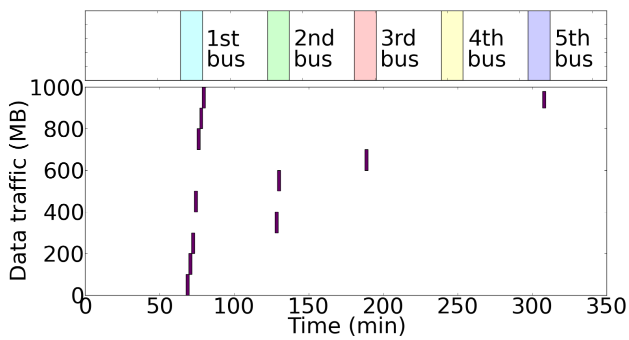

Fig. 13: Data traffic at receiver. 1 GB movie transfer with 100 MB bundle size and random distribution of bundles by the sender.

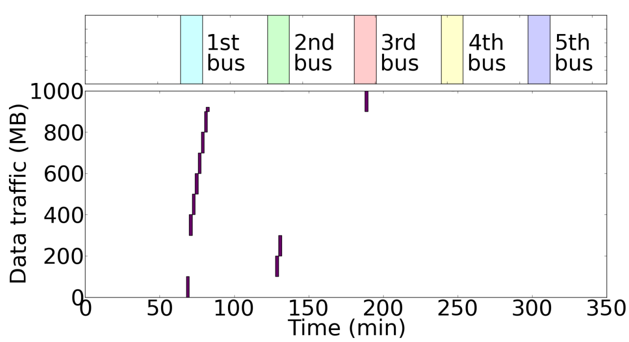

Fig. 14: Data traffic at receiver. 1 GB movie transfer with 100 MB bundle size and FIFO distribution of bundles by the sender.

experimentation process in the laboratory and builds upon the results produced from [5] and [6].

Even though the majority of the tests in DTN related work have been performed on real testbeds or using simulators, efforts have been taken to create realistic emulators for mobile scenarios. RAMON [18] is an emulator that tries to imitate realistic wireless characteristics in high speed systems. A moving node is emulated using an access point, a computer, an attenuator and an antenna. Speed is emulated by changing the signal strength according to a path loss equation. RAMON also uses a propagation model tailored towards indoor and micro-cellular environments. While RAMON is useful in the emulation of the physical layer, it does not scale well. New models must be created for every scenario and attenuators must be placed strategically for the most accurate readings.

An on/off based emulation approach is followed in [19]. The authors claim that a tight interaction between the application and the MAC/PHY layers is missing in existing emulation methods. Their idea is to have a grid of nodes running emulation clients. An emulation server triggers operations. Mobility is emulated by migrating a running application from one node to the other based on a mobility pattern. In other words, on obtaining a trigger from the emulation server, the emulation client on the node would take a snapshot of the state of the running application on the node, stop it and migrate it to the next node on the path, where the application would be restarted. This method satisfies the interaction criterion, but migration of the application introduces latency and the emulation method does not work well in high speed scenarios, where fast migrations would be required. Besides, the size of the snapshots impacts on the performance of the system as it creates further delay. Finally, channel quality fluctuations cannot be emulated by this method.

A large scale spatial switching emulation method is introduced by Ramachandran et al. [20] to tackle large scale topologies. They make use of the m-ORBIT grid testbed and a split-stack architecture to emulate mobility. The application and network layers of a node run on a single machine, the virtual node. The physical and link layers of the grid nodes are used to emulate mobility. Communication between the virtual and grid nodes occurred via tunneling over gigabit Ethernet. The virtual nodes keep on changing tunnels to different grid nodes based on the mobility patterns. Communication between two grid nodes representing two different mobile nodes is done over the wireless interface. Additive white Gaussian noise is introduced into the testbed to disconnect nodes despite their close proximity. This approach is able to emulate mobility on a larger scale and has a much simpler noise model. However, it could not handle high speed and the tunneling introduces some amount of latency. [18] and [19] employs complex designs to emulate mobility in the physical layer. [20], while using a simpler noise model, makes use of tunneling. These decisions resulted in scalability issues. Our design circumvents these issues by forgoing the physical layer emulation and utilizing a less complex emulation method.

With regards to content dissemination, other projects [21], [22], and [23] propose solutions to adapt the BitTorrent protocol to ad-hoc networks. [24] develops a content dissemination protocol for vehicular ad-hoc networks called SPAWN. This protocol is based on a strategy known as rarest-closest first, where a content piece's distribution is determined by a neighboring node's proximity and possession of that piece. [25] follows a similar approach for opportunistic networks, where pieces were disseminated based on their prevalence in the network. The METhoD framework can be used to experiment with strategies based on this prior work and comparisons can be made with our current distribution methods.

VII. Conclusion

The project's overall objective is to deploy a DTN network for media-content distribution in rural South Africa to unleash

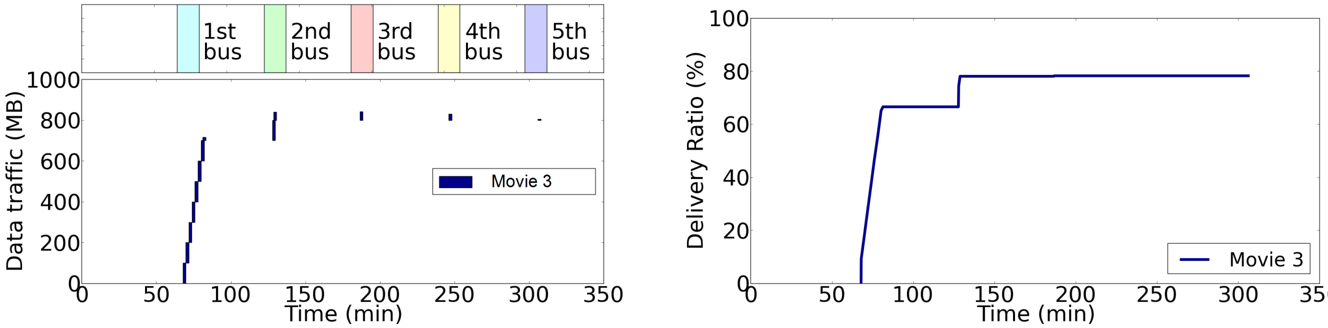

Fig. 15: (Left) Data traffic at receiver. Three 1 GB movies, 100 MB proactive fragmentation, 1-hop routing at receiver. No bundle scheduling. Bundles of only one movie are sent to the receiver by the mobile node. (Right) Delivery ratio.

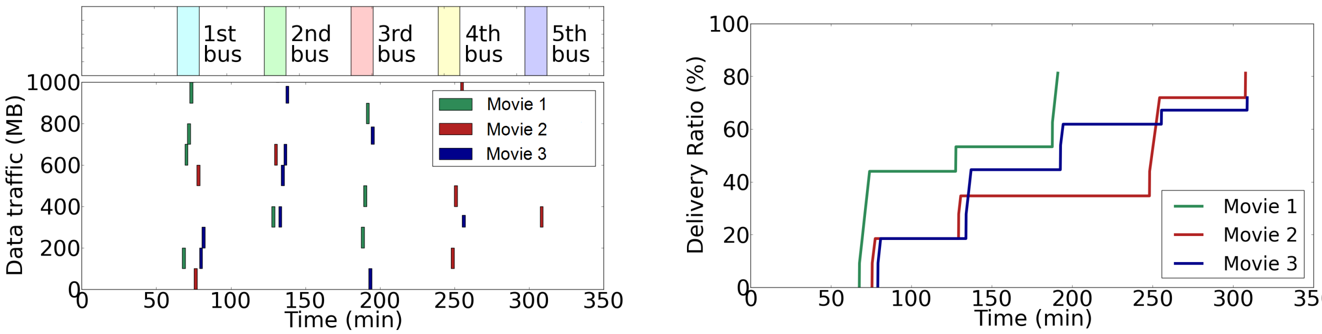

Fig. 16: (Left) Data traffic at receiver. Three 1 GB movies, 100 MB proactive fragmentation, 1-hop routing at receiver. Random scheduling. Bundles of different movies are now sent to the receiver by the mobile node. (Right) Delivery ratio.

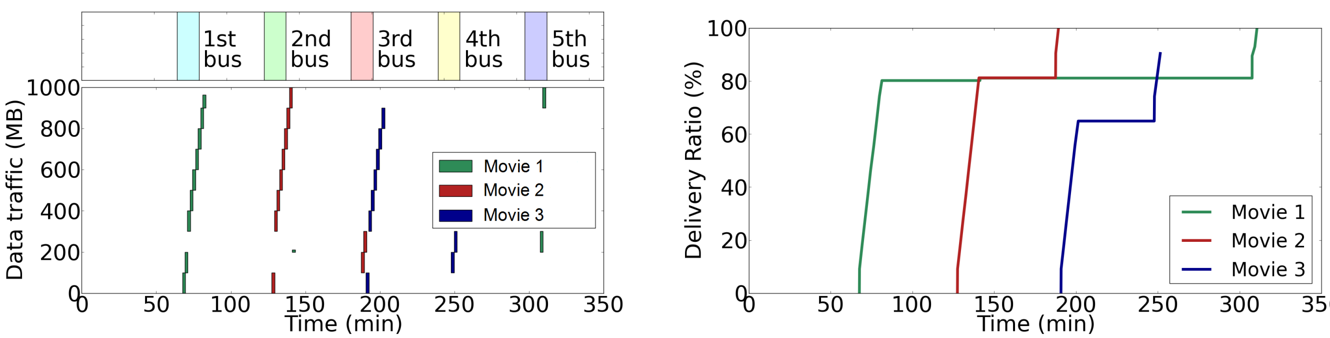

Fig. 17: (Left) Data traffic at receiver. Three 1 GB movies, 100 MB proactive fragmentation, 1-hop routing at receiver. FIFO scheduling. Bundles of different movies are now sent to the receiver by the mobile node. (Right) Delivery ratio.

opportunities for micro-entrepreneurs in such regions. However, before the actual deployment the entire system must be evaluated. To have a convenient but realistic enough method to perform experiments, we have developed METhoD, a mobility emulator testbed for DTNs. METhoD can emulate the mobility of the devices and their connections on the DTN layer. Using our testbed we aim to analyze network performance and identify possible issues before the actual deployment in South Africa. The experiments we have conducted provide interesting insights on the behavior of the devices and the whole network. Analyzing the results of the experiments, we can develop strategies to improve the efficiency of data transfer. We observe that setting up proactive fragmentation at the sender side leads to more efficient use of the contact duration between two nodes. Besides, since bundles are always sent in the same order from the sender, an increase in the number of mobile nodes has no positive effect on the amount of data delivered. A bundle distribution strategy is necessary for higher delivery ratios. A FIFO distribution of bundles is a better option than a random strategy. Furthermore, the METhoD testbed can be used to test different devices in different network scenarios. Finally, the experiments show that the devices have a limitation when it comes to large number of processes and we must ensure that unnecessary processes are not running in the background.

ACKNOWLEDGEMENT

The MOSAIC-2B project is partially funded by the EU FP7 framework program for research, technological development and demonstration under grant agreement no.611796. We thank Johannes Morgenroth and the team of Lars Wolf at Techn. Univ. Braunschweig, Germany, for provisioning software and support of their IBR-DTN. We also thank the Disney Lab Artists, Maurizio Nitti and Alessia Marra, for providing artworks.

REFERENCES

[1] The World in 2014, ICT Facts and Figures. *ITU Technical report, Geneva, Switzerland*, 2014.

[2] Enabling Network Connectivity in Developing Nations. *NYUAD Research Report*, 2012.

[3] K. Scott and S. Burleigh. Bundle Protocol Specification. RFC 5050 (Experimental), November 2007.

[4] MOSAIC 2B: Mobile Empowerment. http://mobile-empowerment.org. Retrieved 3/2015.

[5] S. Siby A. Galati, T. Bourchas and S. Mangold. System Architecture for Delay Tolerant Media Distribution for Rural South Africa. *in the proceedings of the 9th ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization (WiNTECH 2014)*, 2014.

[6] S. Siby S. Frey M. Olivares A. Galati, T. Bourchas and S. Mangold. Mobile-Enabled Delay Tolerant Networking in Rural Developing Regions. *in the proceedings of the IEEE Global Humanitarian Technology Conference (GHTC 2014)*, 2014.

[7] A.Vahdat, D.Becker, et al. Epidemic routing for partially connected ad hoc networks. Technical report, Technical Report CS-200006, Duke University, 2000.

[8] J.Burgess, B.Gallagher, D.Jensen, and B.N.Levine. Maxprop: Routing for vehicle-based disruption-tolerant networks. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1–11, April 2006.

[9] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott. Impact of human mobility on opportunistic forwarding algorithms. *IEEE Trans. Mob. Comput.*, 6:606–620, 2007.

[10] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott. Pocket switched networks: Real-world mobility and its consequences for opportunistic forwarding. *Technical Report UCAM-CL-TR-617, University of Cambridge*, February 2005.

[11] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot. Pocket switched networks and human mobility in conference environments. *In Proceedings of the ACM SIGCOMM*, 2005.

[12] J. Leguay, A. Lindgren, J. Scott, T. Friedman, and J. Crowcroft. Opportunistic content distribution in an urban setting. *Proceedings of the 2006 SIGCOMM workshop on Challenged networks, CHANTS '06*, pages 205–212, 2006.

[13] L Vanajakshi, SC Subramanian, and R Sivanandan. Travel time prediction under heterogeneous traffic conditions using global positioning system data from buses. *IET intelligent transport systems*, 3(1):1–9, 2009.

[14] Julien Montavont and Thomas Noel. Ieee 802.11 handovers assisted by gps information. In *Wireless and Mobile Computing, Networking and Communications, 2006.(WiMob'2006). IEEE International Conference on*, pages 166–172. IEEE, 2006.

[15] Tommy Nguyen and Boleslaw K Szymanski. Using location-based social networks to validate human mobility and relationships models. In *Advances in Social Networks Analysis and Mining (ASONAM), 2012 IEEE/ACM International Conference on*, pages 1215–1221. IEEE, 2012.

[16] W. Pottner S. Schildt, J. Morgenroth and L. Wolf. IBR-DTN: A lightweight, modular and highly portable Bundle Protocol implementation. *Electronic Communications of the EASST*, pages 1–11, 2011.

[17] A modular and lightweight implementation of the bundle protocol. http://trac.ibr.cs.tu-bs.de. Retrieved 2/2015.

[18] E.Hernandez and A.S.Helal. RAMON: Rapid-Mobility Network Emulator. In *Proceedings of the 27th Annual IEEE Conference on Local Computer Networks*, LCN '02, pages 809–820, Washington, DC, USA, 2002. IEEE Computer Society.

[19] H.Yoon, J.Kim, M.Ott, and T.Rakotoarivelo. Mobility emulator for dtn and manet applications. In *Proceedings of the 4th ACM International Workshop on Experimental Evaluation and Characterization*, WINTECH '09, pages 51–58, New York, NY, USA, 2009. ACM.

[20] K.Ramachandran, S.Kaul, S.Mathur, M.Gruteser, and I.Seskar. Towards large-scale mobile network emulation through spatial switching on a wireless grid. In *Proceedings of the 2005 ACM SIGCOMM Workshop on Experimental Approaches to Wireless Network Design and Analysis*, E-WIND '05, pages 46–51, New York, NY, USA, 2005. ACM.

[21] Sundaram Rajagopalan and Chien-Chung Shen. A cross-layer decentralized bittorrent for mobile ad hoc networks. In *Mobile and Ubiquitous Systems: Networking & Services, 2006 Third Annual International Conference on*, pages 1–10. IEEE, 2006.

[22] Neelakantam Gaddam and Anupama Potluri. Study of bittorrent for file sharing in ad hoc networks. In *Wireless Communication and Sensor Networks (WCSN), 2009 Fifth IEEE Conference on*, pages 1–6. IEEE, 2009.

[23] Mohamed Karim Sbai, Emna Salhi, and Chadi Barakat. P2p content sharing in spontaneous multi-hop wireless networks. In *Communication Systems and Networks (COMSNETS), 2010 Second International Conference on*, pages 1–10. IEEE, 2010.

[24] Alok Nandan, Shirshanka Das, Giovanni Pau, Mario Gerla, and MY Sanadidi. Co-operative downloading in vehicular ad-hoc wireless networks. In *Wireless On-demand Network Systems and Services, 2005. WONS 2005. Second Annual Conference on*, pages 32–41. IEEE, 2005.

[25] N.Belblidia, M.Dias de Amorim, L.H.M.K.Costa, J.Leguay, and V.Conan. Pacs: Chopping and shuffling large contents for faster opportunistic dissemination. In *Wireless On-Demand Network Systems and Services (WONS), 2011 Eighth International Conference on*, pages 9–16, Jan 2011.