TECHNISCHE UNIVERSITÄT DRESDEN

NETWORKING GROUP

FIRST TAB REPORT

Role based DTN routing adaptation

Author:
José IRIGON

Supervisors:
Prof. Dr. rer. nat. habil. Dr. h. c. Alexander SCHILL
Prof. Dr. Thorsten STRUFE
Dr.-Ing. Thomas SPRINGER

November 22, 2018

Abstract

1 Introduction

The Internet of Things has grown immensely over the last years. Although most current solutions are based on traditional Internet protocols (and therefore require end-to-end connectivity), a great amount of opportunities does not fulfill this requirement and is yet to be explored.

Disruption Tolerant Networking (DTN) aims those opportunities, but its adoption has been much lower than traditional protocols. In this work, the state of the art in DTN is present as well as the contribution we expect to accomplish towards the improvement in DTN adoption over the next two and a half years.

This document is organized as follows: In section 2, the DTN background, in which DTN network priorities and challenges regarding routing and congestion control are discussed, will be presented. Section 3 explains why adaptability is necessary in DTN, followed by a concrete example of how the concept of roles could provide the dynamic adaptation in section 4. Section 5 concludes this work with a time plan showing my progress so far.

2 DTN background

Disruption Tolerant Networking handle the challenges of exchanging data in a heterogeneous network that lacks end-to-end connectivity. Regardless of the efforts put into the subject in the last 15 years, areas as routing algorithms, congestion control, fairness and QoS are still under active research.

In this work, we are specially interested in a subset of the DTN world in which a network is composed by static communicating devices called stations and mobile communicating devices used as data mules. Below, a brief overview about some well known DTN use-cases is presented:

1. The village case[1, 2, 3, 4, 5, 6, 7]: In places where the Internet infrastructure is not economically viable, trains or buses are used as data mules to carry information between stations in a latency-insensitive communication. Examples are Daknet, Trainnet, Kiosknet, Village, etc.

2. Disaster Recovery[8, 9, 10]: After natural catastrophes the infrastructure is vanished. There is a need for fast-deployed, dynamic, ad-hoc networks taking advantage of the efforts and resources provided by survivors and first-aid groups. In such cases, ambulances and people are data mules to exchange data between points of data exchange, where the data is consumed and decisions are taken.

3. Ring-Road [11]: This approach uses micro satellites as Cubesats built on custom hardware to provide latency insensitive Internet services using LEO (Low Earth Orbit) satellites as data mules to provide communication between ground stations.

4. Smart Farming[12]: Large areas of grass can be monitored using low range motes that act as static stations and communicate with animals (cows, horses) that move the sensed data back to a main station to be processed, allowing for a low-budget sampling of the whole area.

A generic example of a DTN network topology is depicted in Figure 1, where static stations are represented by squares. They can be used as strategic data exchanging points to forward data or serving basically as a post box to clients of this network. The coloured lines show the possible circuits travelled by mobile nodes (not depicted in the figure) in this network, similar to bus or train lines.

2.1 The priorities of a DTN

Nodes in a DTN network may different objectives that might be conflicting. For example, if nodes in the network run on battery, it might be important to minimize replications (and therefore transmissions) and assure a fair distribution of load; otherwise the most active nodes will be the first to run out of battery. Other use-cases may have similarities with real-time systems being only interested in the last state of an information and try to minimize the delivery latency. Finally, in some cases the application is latency-insensitive, but the user is interested in minimizing the information loss. Below, we list some common targets of DTN:

- Maximize delivery probability
- Maximize energy efficiency

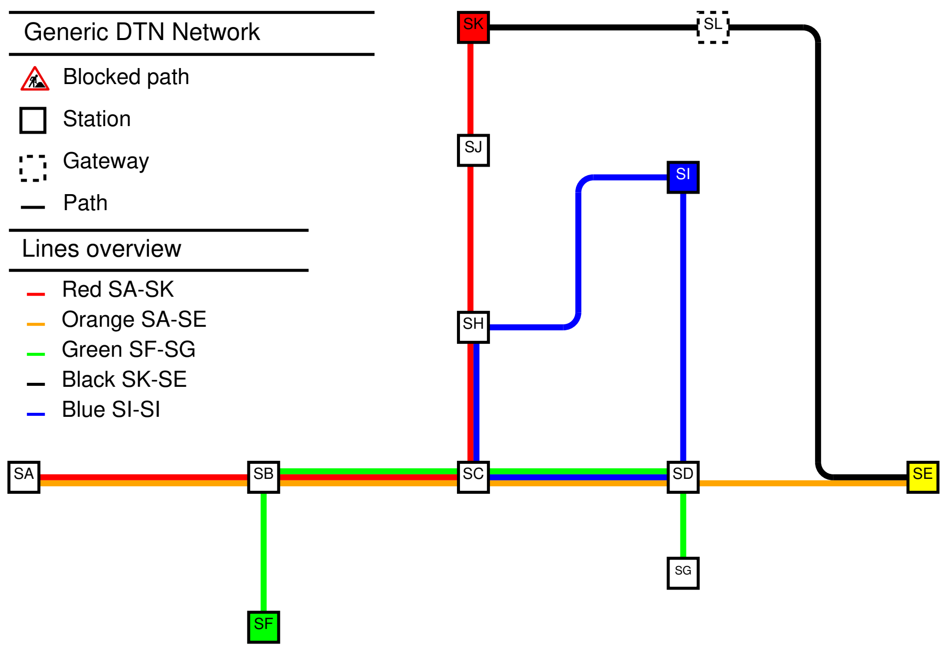


Figure 1: Opportunistic Data-mule Network

- Minimize end-to-end latency
- Minimize the number of hops
- Minimize queuing delay

In order to achieve its desired targets, a DTN node should choose the routing algorithm that based on the given contextual information moves data towards its destination while making efficient use of its resource, and therefore avoiding congestion.

Note that such decisions, in contrast to traditional protocols, must be taken locally based on the current (possibly outdated) view of the network since connectivity between nodes within this network can not be assumed at any time.

2.2 Routing in DTN

When a node in a DTN network has information to send or forward, it has to perform the **routing decision**, i.e it must decide to which neighbor(s) and from which point in time the forwarding may happen.

Over the last years several routing algorithms were proposed. We summarize them briefly in three groups:

1. **Zero-knowledge**: protocols in this group decide whether or not to replicate the data in a contact opportunity regardless of historic or statistical events. They have the simplest implementations and may be further divided in single-copy (e.g Direct Delivery[13]) or naive-replication (e.g Epidemic[14] and binary Spray Wait[15]). Naive-replication routing is considered to be resource-expensive (due to flooding behavior), present low end-to-end transmission latency under low load (all paths including the shortest one are travelled) and bad performance (high latency, bundle lost, low delivery probability) when the network load or density increases.

2. **Probability-based**: Protocols in this category use historic information and derive predictions (e.g the expected probability of a device dA to meet a device dB directly or transitively in the future). Based on such predictions, routing algorithms are expected to choose a subset of next hop candidates from all possible neighbors, maximizing the probability of bundle delivery respecting the priority described above in 2.1. Examples of probability-based DTN routing algorithms are ProPHET[16], Spray Focus[17] and MH* [18].

3. **Schedule-based**: When the movement of the mobile devices are known in advance (e.g in satellite and space communications, contacts can be predicted based on orbital movements) the list of contact opportunities may be given to each node[1] as input. An entry in a contact plan may contain, for example, the start, duration and maximum capacity of each future encounter between every pair of nodes. When a contact plan is available, the routing decision can be translated in finding out the next hop in the shortest path using, for example, Dijkstra. An example of scheduled-based algorithm for DTN is Contact Graph Routing[19] (CGR). Although scheduled-based algorithms are in the literature mainly applied in space communications, several terrestrial DTN use-cases (see the "village case" class in section 2) may follow predefined schedules. Finally, schedule based networks tend to present the best performance among all strategies (when the schedule is respected)

[1] Any device in a DTN network

due two reasons: first, it is able to make the most precise predictions. Second, it may take capacity in account. Still, congestion even in such cases are an open challenge, since the routing decision is made locally. After a bundle is forward to the next hop (based on the best path calculated locally), the next will be free to decide how to forward based on its understanding of the network.

2.3 Congestion Control in DTN

According to [20], congestion occurs when resource demand exceeds networks capacity. If the buffer of a node is full at the moment a neighbor asks to forward a bundle, the request has to be either denied or the node must make space deleting bundles from its own buffer. Another resource that a node may become short of is transmission capacity. The amount of data transmitted within a contact opportunity depends not only on the contact size and transceivers capability, but also in the density of the network and how many neighbors in the vicinity may share the channel when the contact occurs.

There is currently no definitive congestion control algorithm for DTN, and this is an area of research on its own. Some routing algorithms try to embed some measures to minimize the changes of a congestion happen; congestion control mechanisms may also act by means of queue policies dropping bundles selectively or deciding the order in which bundles should be forward. In [20] the author suggests that congestion recognition is done basically based on network capacity, buffer availability and drop rate .

In this work **we focus exclusively on congestion control mechanisms embedded on the routing decision**, basically constraining the amount of replication based on the sensed context of the network. Congestion control algorithms can classified as pro-active, reactive or hybrid, depending on the point in time that the algorithm should act. Regarding network feedback they can be alternatively classified as open- (no back channel used to verify congestion or control effectiveness) or closed-loop (network is sensed to recognize congestion and the effects of congestion control measures). Some are routing specific and others may act independent of the routing protocol.

Examples of congestion controls used in this work are:

1. The retiring replicants[21]: This hybrid approach reacts to congestion controlling the number of copies a node may replicate in an encounter.

2. Using buffer space advertisements[22]: Through buffer space advertisements, next hops that contains high occupancy are avoided whenever possible.

2.4 Achieving DTN targets

In a DTN there may be data exchange between a set of possible resource-constrained peers. If the amount of data to be exchanged is small compared to the transmission and storage capabilities, replicating the data at each encounter (flooding) can be an effective approach that transfers the data at least latency. This approach is, however, ineffective in case the amount of resource (storage, transmission capacity) is limited or in case the network density is high.

The idea of this work is to combine routing algorithms with congestion avoidance or control depending on the context sensed. Under context change, both routing algorithm and congestion management strategy should be adapted.

3 The need for adaptability in DTN

To the best of our knowledge, there is no use case in which the routing algorithm is adapted during run time. The DTN topology may however change drastically. Take for example a disruption tolerant network based on trains that are used as data mules to move data between train stations. Trains and stations may have a contact plan that allows them to calculate the best neighbor to forward a message based on the shortest path (Dijkstra).

Due to a storm, a bridge (in the train path) may be damaged, deviating the normal route of the train for a while. From the point of view of its routing module, stations that should not be on the way are met. At this point, it is clear that it has an invalid contact plan, so that it should either forward the messages to destinations that are on the way (direct contact routing) or forward to every station that comes in contact (epidemic routing).

As time goes by, the train may repeat the new path several times a day, such that after some point the train could start routing probabilistically (based on the information it gets from its contacts).

In our work, we expect to be able to show that an algorithm, which is able to adapt under such event, should outperform algorithms that use

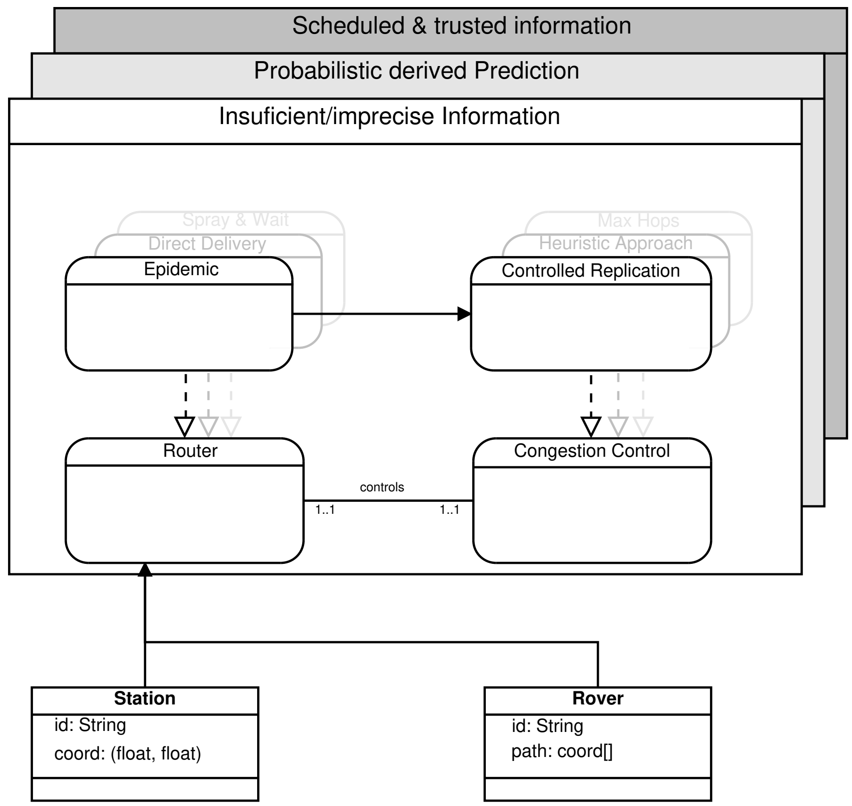


Figure 2: Possible role-based approach

either deterministic routing based on a contact plan (that becomes invalid after some time) or epidemic during the whole execution.

4 Using the model of roles to provide DTN adaptability

In this section, we will present how the concept of roles and the tools provided by RoSI can be incorporated in order to support the design, modelling and implementation of a role-based framework for adaptive routing.

The concept of roles allow behavior to be acquire or dropped during the execution (player plays role), such that unrelated objects may become related in run time. In the network presented above, static nodes (the train stations) and mobile nodes (trains) can be modeled as players, while routing decision and congestion management decisions as roles.

When an object (e.g. train) receives or creates a message, it has to decide to which neighbors it should be forwarded and possibly bound the maximum number of replicas that may be created. In the simplest case, no information about the context is taken in account and the train might decide to take the routing decision based on an epidemic approach (flooding) as depicted in figure 2.

After the routing decision is taken, deciding upon the set of next-hop candidates (in the case of Epidemic routing is any), a congestion control mechanism may be chosen to control replication. Lets assume that, from the available congestion control mechanisms, two may be used with Epidemic routing, leaving the system with the following possibilities:

- Controlled Replication: set the maximum number of replicas based on the neighbor buffer availability

- Max Hops: only forward messages if the number of hops is less than a specific number

- null: do not choose a congestion control, or set the number of replicas to infinity.

Based on the context sensed by the node, it will decide which is the best mechanism for the time being.

As the time passes, the train may collect additional context information about the network transitively. At each contact, trains and stations may exchange information about their knowledge regarding the network topology[2].

Based on the information collected on each contact, the device may notice that contacts happen in a rather predictable form. At the event of a new message, the device notice based on the context that enough information is available and set the compartment to "Probabilistic derived Prediction", allowing the device choose one among the set of possible probabilistic routing protocols (e.g. ProPHET), which in turn would have a set of one or more congestion management mechanism that together try to achieve the network

[2] DTNs are expected to be reasonably small. Algorithms as ProPHET expects explicitly a size up to 30 nodes

targets avoiding wasting resources (unnecessary replication) whenever possible.

5 Timeplan

In this section, the achievements and future work is presented. Figure 3 gives a broad overview of what is being done so far and how is the work organized.

5.1 Related work

There are three areas I am focusing my work. The concept of roles has been improved through RoSI lectures, workshops and discussions. However, there are still role concepts (e.g internal roles, group constrains, etc.) and tools (e.g CROM and FramED) to be learned.

We are currently studying the influences that routing algorithms and congestion control mechanisms practice on each other. A classification will be necessary selecting groups of routing algorithms and congestion control mechanisms of interest to our work as well as define the scope of the thesis based on it.

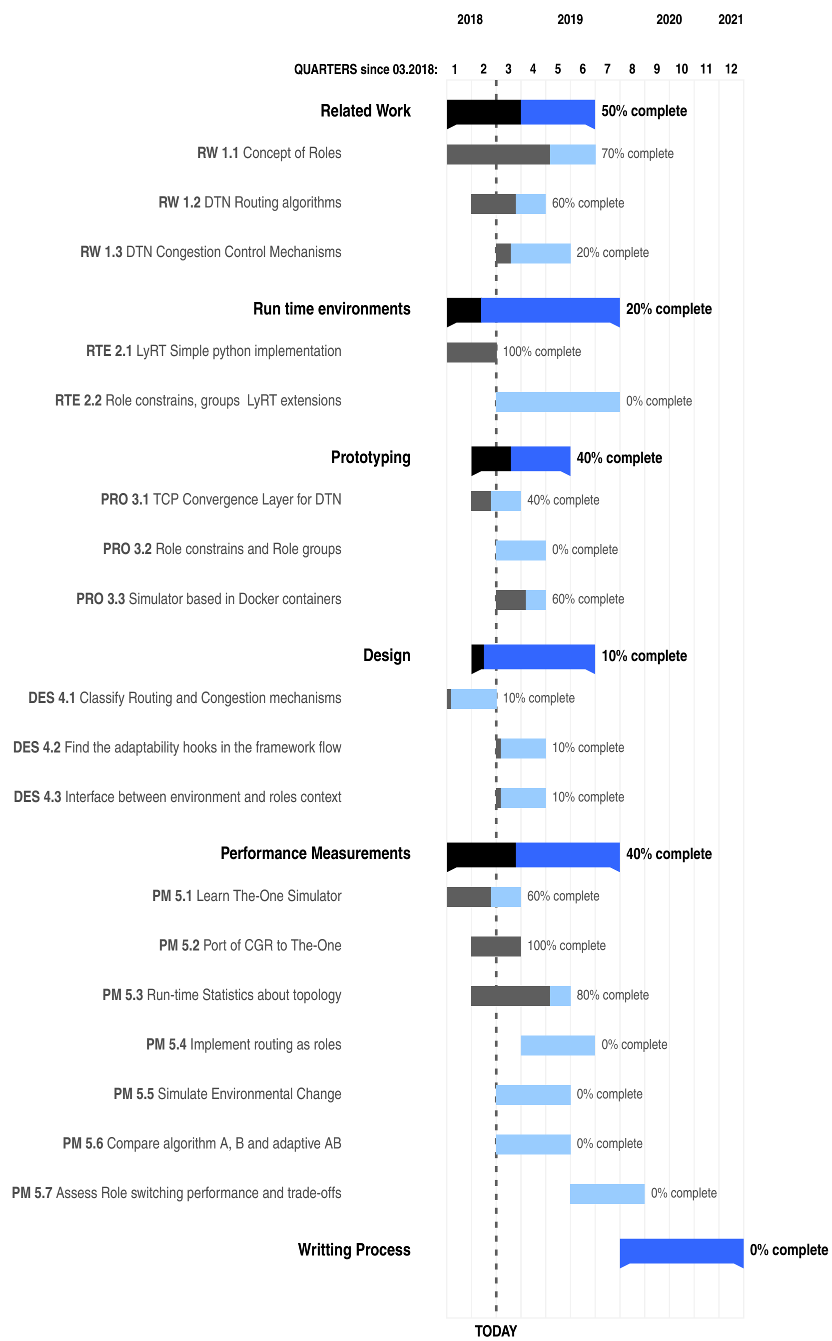
5.2 Run time environments

For this work, a simple implementation of LyRT was implemented to verify that the role concepts were properly understood and to have an idea about the trade-offs using a role-based run time environment (pyLyRT is available at [23]). The framework will need to be further extended, adding features as role constrains and role groups. Therefore we need must verify if there is any framework (e.g SCROLL) that have such features embedded or discuss with RoSI coleagues about the possible trade-off on the implementation of such features.

5.3 Prototyping

It is still unclear if our results will be compared by means of simulation or real prototypes. In the first months of this research, we implemented a convergence layer that should sit on the top of TCP for two DTN nodes to communicate with each other.

Figure 3: Time line overview



Besides, a simple mechanism that manage connection between nodes was also implemented. Based on a contact plan, a POX controller set the state (forward or drop) of a SDN switch port deciding when there should be connectivity between each pair of peers in the network. This mechanism may be used to simulate connection state in a DTN network.

5.4 Design

So far we invested less time on the design of the framework, focusing primarily in finding out in which scenarios adaptation may happen and adaptive protocols are needed.

5.5 Performance Measurements

This section summarizes efforts referring to simulations done so far and the assessments we plan to do. Simulations have been performed with the Opportunistic Network Simulator [13] (The-One). A CGR implementation was added to the simulator as well as a run time graph that can be created in run time to collect statistics about the DTN network. Based on those statistics we expect to quantify the context and understand if a role adaptation needs to be triggered.

Acknowledgment

This work is funded by the German Research Foundation (DFG) within the Research Training Group Role-based Software Infrastructures for continuous-context-sensitive Systems (GRK 1907).

References

[1] A. Lindgren and A. Doria, "Experiences from deploying a real-life DTN system," *2007 4th Annual IEEE Consumer Communications and Networking Conference, CCNC 2007*, pp. 217–221, 2007.

[2] K. Heimerl and E. Brewer, "The village base station," *Proceedings of the 4th ACM Workshop on Networked Systems for Develop-*

ing Regions - NSDR '10, pp. 1–2, 2010. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1836001.1836015

[3] A. Lindgren, A. Doria, J. Lindblom, and M. Ek, "Networking in the land of northern lights," *Proceedings of the 2008 ACM workshop on Wireless networks and systems for developing regions - WiNS-DR '08*, pp. 1–7, 2008. [Online]. Available: http://dl.acm.org/citation.cfm?id=1410064.1410066

[4] S. Grasic, *Development and Deployment of Delay Tolerant Networks: An Arctic Village Case*, 2014. [Online]. Available: https://www.diva-portal.org/smash/get/diva2:989908/FULLTEXT01.pdf

[5] S. Grasic and A. Lindgren, "Revisiting a remote village scenario and its DTN routing objective," *Computer Communications*, vol. 48, pp. 133–140, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.comcom.2014.04.003

[6] A. Pentland, R. Fletcher, and A. Hasson, "DakNet: Rethinking Connectivity in Developing Nations," *Computer*, vol. 37, no. 1, 2004.

[7] M. Zarafshan-Araki and K. W. Chin, "TrainNet: A transport system for delivering non real-time data," *Computer Communications*, vol. 33, no. 15, pp. 1850–1863, 2010. [Online]. Available: http://dx.doi.org/10.1016/j.comcom.2010.06.008

[8] G. Tuna, B. Nefzi, and G. Conte, "Unmanned aerial vehicle-aided communications system for disaster recovery," *Journal of Network and Computer Applications*, vol. 41, no. 1, pp. 27–36, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.jnca.2013.10.002

[9] K. Kannan, F. Silveira, and N. Taft, "A Disruption-Tolerant Architecture for Secure and Efficient Disaster Response Communications," *Kfall.Com*, 2008. [Online]. Available: http://kfall.com/ucbpage/papers/dtn-iscram.pdf

[10] C. B. Nelson, B. D. Steckler, and J. A. Stamberger, "The evolution of hastily formed networks for disaster response: Technologies, case studies, and future trends," *Proceedings - 2011 IEEE Global Humanitarian Technology Conference, GHTC 2011*, no. Dmi, pp. 467–475, 2011.

[11] S. C. Burleigh and E. J. Birrane, "Toward a communications satellite network for humanitarian relief," *Proceedings of the 1st International Conference on Wireless Technologies for Humanitarian Relief - ACWR '11*, p. 219, 2011. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2185216.2185280

[12] C. Kulatunga, L. Shalloo, W. Donnelly, E. Robson, and S. Ivanov, "Opportunistic Wireless Networking for Smart Dairy Farming," *IT Professional*, vol. 19, no. 2, pp. 16–23, 2017.

[13] A. Keränen, J. Ott, and T. Kärkkäinen, "The ONE simulator for DTN protocol evaluation," *Proceedings of the Second International ICST Conference on Simulation Tools and Techniques*, 2009. [Online]. Available: http://eudl.eu/doi/10.4108/ICST.SIMUTOOLS2009.5674

[14] A. Vahdat and D. Becker, "Epidemic Routing for Partially-Connected Ad Hoc Networks."

[15] C. S. Raghavendra, "Spray and Wait : An Efficient Routing Scheme for."

[16] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," vol. 7, no. 3, p. 19, 2003. [Online]. Available: http://portal.acm.org/citation.cfm?doid=961268.961272

[17] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: The multiple-copy case," *IEEE/ACM Transactions on Networking*, vol. 16, no. 1, pp. 77–90, 2008.

[18] V. Conan, J. Leguay, and T. Friedman, "Fixed point opportunistic routing in delay tolerant networks," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 5, pp. 773–782, 2008.

[19] S. Burleigh, C. Caini, J. J. Messina, and M. Rodolfi, "Toward a Unified Routing Framework for Delay- Tolerant Networking," *IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE)*, pp. 82–86, 2016.

[20] A. P. Silva, S. Burleigh, C. M. Hirata, and K. Obraczka, "A survey on congestion control for delay and disruption tolerant networks," *Ad*

Hoc Networks, vol. 25, no. PB, pp. 480–494, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.adhoc.2014.07.032

[21] N. Thompson, S. C. Nelson, M. Bakht, T. Abdelzaher, and R. Kravets, "Retiring replicants: Congestion control for intermittently-connected networks," *Proceedings - IEEE INFOCOM*, 2010.

[22] J. Lakkakorpi, M. Pitkänen, and J. Ott, "Using buffer space advertisements to avoid congestion in mobile opportunistic DTNs," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6649 LNCS, pp. 386–397, 2011.

[23] J. Irigon, "A lyrt python framework," Nov. 2018. [Online]. Available: https://doi.org/10.5281/zenodo.1494119