Experiments and results on DTN for IOT III Urbanet collaboration

Patrice Raveneau, Hervé Rivano

Experiments and results on DTN for IOT
III Urbanet collaboration

Patrice Raveneau , Hervé Rivano

ISRN INRIA/RT-472--FR+ENG
ISSN 0249-0803

TECHNICAL REPORT
N° 472
December 2015
Project-Team Urbanet

Experiments and results on DTN for IOT III Urbanet collaboration

Patrice Raveneau , Hervé Rivano

Project-Team Urbanet

Technical Report n° 472 — December 2015 — 15 pages

Abstract: This document presents the full study of the DTN (Delay Tolerant Network) within IOT (Internet Of Things) context. The motivation for using generic protocols able to handle the constraints due to the IOT is highlighted with the choice of the Bundle Protocol. A study of existing implementations of this protocol is realised within a sensor context. We justify the choices made for our implementation, then we define the mechanisms which we test with the Cooja platform by following the protocol of tests we have developed. The results of the experiments are analysed.

Key-words: DTN, IOT, IOT-Lab, Cooja, Contiki

RESEARCH CENTRE GRENOBLE – RHÔNE-ALPES

Inovallée
655 avenue de l'Europe Montbonnot
38334 Saint Ismier Cedex

Expériences et résultats pour les DTN pour l'IOT Collaboration III Urbanet

Résumé : Ce document présente l'étude complète des DTN (Delay Tolerant Network) dans un contexte d'IOT (Internet Of Things). La motivation de l'utilisation de protocoles génériques capables de supporter les contraintes inhérentes au contexte IOT est mise en avant avec le choix du Bundle Protocol. Une étude des implantations existantes de ce protocole est faite dans un contexte capteur. Nous proposons notre implantation en justifiant les choix réalisés, puis nous définissons des mécanismes que nous testons avec la plate-forme Cooja en suivant le protocole de tests que nous avons élaboré. Les resultats de ces expérimentations sont analysées.

Mots-clés : DTN, IOT, IOT-Lab, Cooja, Contiki

1 Introduction

The Internet Of Things (IOT) refers to one part of the smart world developping recently where each device, each person and each object is connected through one or several communicating devices and interact with each other [1].

The objective of this document is to define protocol architectures to respond to the heterogeneity that may arise in a smart context with several devices with several mobility behaviours. We also define a test scenario to analyse the performance of a network using our proposal in an IOT context. For an IOT scenario, there is a lot of data communications but generated by small data volumes such as a measurement of temperature, air quality or parking occupancy.

We describe the context of using sensors and raise the difficulties of this context. Subsequently, we propose a protocol architecture fitting the requirements of several IOT scenarios before considering the implementations constraints on small devices. Once this framework defined, we propose mechanisms able to use mobility of devices to reduce the number of useless communications in order to better use network capacity. Eventually, we define the tests scenario to validate our proposals and analyse the performance of the network.

2 Context

The sensors will be considered static or mobile as required by the intended IOT application. In the case of static nodes, their distribution may be regular or random depending on the conditions of the deployment. However, the literature is abundant on static sensor networks and addresses several problems. The major problems are related to energy management [2]. The solutions proposed to solve this problem based on routing protocols and algorithms using the geographical distribution of nodes [3]. Load distribution techniques in the network [4] and selecting routing trees minimizing the period of data collection [5] allow also to reduce the energy consumption. Energy can also be saved by using an optimal transmission range [6] and cutting the network into a number of optimal groups [7].

Sensor networks deployed within a city might count thousands of nodes. Nevertheless, for our tests, we will select key elements to represent the several nodes behaviours that might occur in such a network.

For our study, we consider static and mobile sensors communicating to a single sink through one or several mobile gateways. We consider static sensors to monitor environment and mobile sensors which are able to collect data from the static ones. We use mobile sensors as relays between two unconnected sensors segments of a sensor network.

The information exchanged over the network are control messages of data. Included in the control signaling for determining routes between nodes and data acknowledgments.

3 Architecture protocol

We now present the protocols used within the network.

We highlight the main constraints of our network, the heterogeneous technologies from urban environment, the lack of connectivity, the dynamic mobility and long links interruptions. Protocols like Internet Protocol (IP) are not able to manage these constraints. If a link is unavailable, then the datagram is lost. A reliable protocol such as Transmission Control Protocol (TCP) will handle retransmissions. But the duration of such interruptions are too large to allow TCP to be effective. That's why we decided to focus on Delay/Disruption Tolerant Networks (DTN).

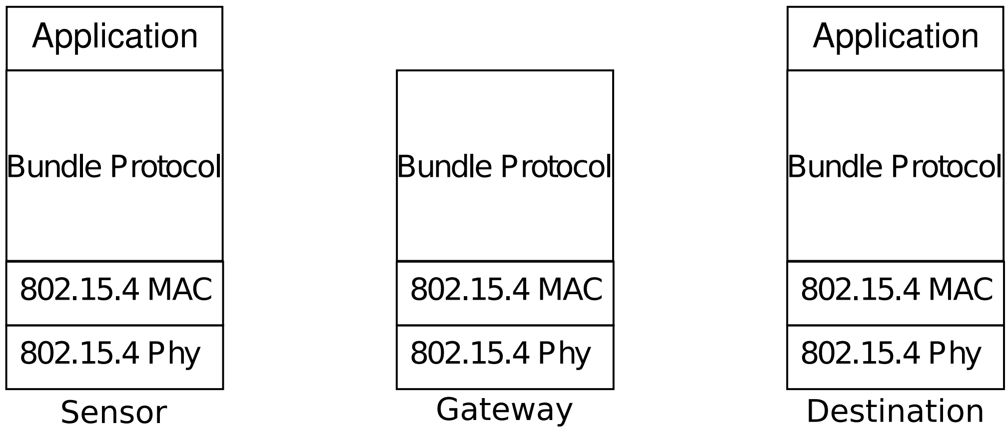

Figure 1: Architecture with Bundle Protocol

The Bundle Protocol which meets the requirements of DTN architecture is able manage the key constraints. In addition, it allows to interconnect networks with different protocol architectures.

3.1 A DTN architecture

For sensors in an IOT purpose, it is necessary to limit the number of protocols layers to minimize the use of memory resources. Thus we arrived at the definition of the protocol architecture in figure 1 .

On the protocol architecture we defined, the Bundle Protocol is implemented on each system, even on sensors. Sensors are able to store data they do not have themselves created. We clarified that, given the small memory sensors, we had to limit the number of protocols used on that equipment. That's why we do not use network layer as IPv6 Low power Wireless Personal Area Networks (6LoWPAN). Indeed, the protocol 6LoWPAN [8] brings nothing more to our architecture and can not manage long discontinuities. The Bundle Protocol actually allows naming all entities in a network. In addition, the fields indicating the source and destination are Self-Delimiting Numeric Value (SDNV). The goal is to reduce the overhead on sensors communications with small headers.

The main drawback of this architecture is that it consumes a part of the sensors memory. Adding a protocol level using several mechanisms such as custody management, storage or reports consumes memory resources. However, in a context where nodes are mobile, it is necessary that the terminal can store data from other sources in order to increase the probability of delivery.

It will be useful to be able to select the best data to share and store. Such a solution makes better use of resources. Similarly, the use of acknowledgment mechanisms increases the memory usage; however it reduces the transmissions of Bundles that have already been delivered to a gateway.

We do not consider the Bundle Protocol as an overlay protocol, linking very different networks, but as a way to overcome the constraints caused by long interruptions and dynamic mobility scenarios .

The use of the Bundle Protocol on the entire network allows to consider gateways as a relay and not as the destination data from the sensors network. Nevertheless, this relay plays a role of custodian and reduces the load in the sensor network indicating that Bundles left this part

Table 1: Advantages and drawbacks of the proposed architecture

Advantages	Drawbacks
Full DTN architecture	Overhead in the network
All equipments interacting	Memory limitation on sensors

of the network. The last node to have transmitted Bundles to the gateway removes it from its memory.

The architecture we have proposed is intended generic. We do not wish to provide architectures that are tailored to a single specific scenario. Our architecture can adapt to other scenarios. The mobile gateway could be replaced by a static one, the proportion of static and mobile nodes could change and the architecture would remain the same.

3.2 Bundle Protocol implementation on sensors

We want to prove that the Bundle Protocol could be implemented on sensors. For this, we studied the existing codes before achieving ours.

3.2.1 ION

The first implementation we present is Interplanetary Overlay Network (ION) developed by Jet Propulsion Laboratory (JPL). This implementation was performed with the goal of being deployed in a context of interplanetary communications. To ensure the robustness of their systems, the missions of the JPL do not use dynamic memory allocation. In addition, for interplanetary missions, it is essential to recover data, even partial, so overhead induced by the Bundle Protocol should be minimal [9]. ION is a complete implementation as it offers adaptation layers with TCP, User Datagram Protocol (UDP) and Licklider Transmission Protocol (LTP). The implementation of LTP is also provided.

While this implementation contains all features and limits the use of dynamic memory mechanisms, it is not appropriate to our sensor context. The use of pre-allocation memory mechanisms is perfect for our purpose. Nevertheless, the existence of various libraries increases the code size and does not meet one of the major constraints implementing on sensor, low memory.

We focus now on the Bundle Protocol reference implementation.

3.2.2 DTN2

DTN2 is the implementation of the Research Group DTN and meets requirements of [10]. This implementation provides several layers including IP and Ethernet convergence layers. Convergence layers to the level 2 and 3 protocols show that it is possible to deploy the Bundle Protocol on low-level protocols and then limit the overhead, which is an important aspect in the IOT context.

This implementation, however, has significant disadvantages for implementation on sensors. Unlike ION, the usage of dynamic memory is not restricted. In addition, the programming language used is C++ which is too resource hungry for deployment on sensors.

We turn our study to implementations thought for embedded systems.

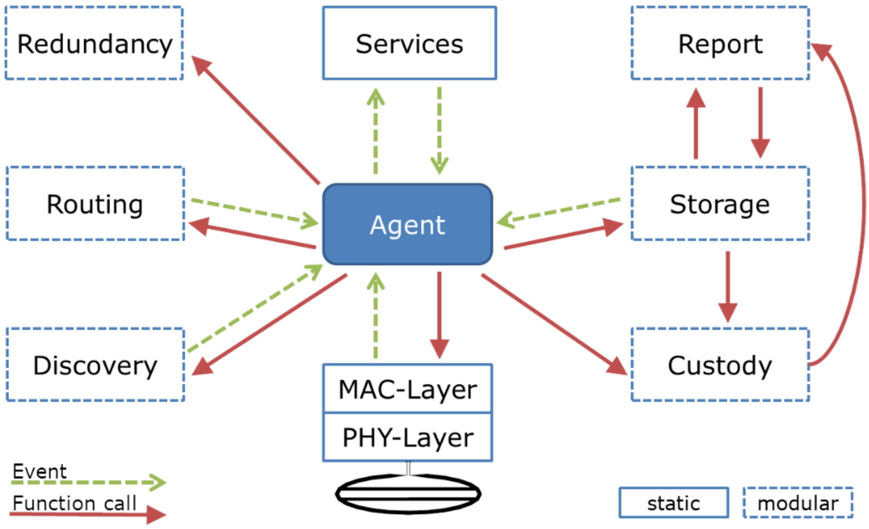

Figure 2: Architecture of μDTN taken from [13]

3.2.3 IBR-DTN

IBR-DTN is an implementation for embedded systems [11]. Unlike other solutions designed for small systems IBR-DTN can work with equipment using other implementations of the Bundle Protocol [12]. In addition to convergence layers on UDP and TCP , this implementation also includes a layer of convergence to the IEEE 802.15.4 protocol. This implementation is adapted to an IOT context.

The problem is that the code size of this implementation is still larger than the 48K that we have at our disposal. We continue our search for implementations that would suit our equipment.

3.2.4 μDTN

μDTN is designed to be suitable for sensor networks [13]. Unlike DTNLite [14] that uses the DTN concept without implementing the Bundle Protocol and ContikiDTN [15] using the Bundle Protocol over a convergence layer on TCP, μDTN works directly above a convergence layer to the IEEE 802.15.4 protocol. We present in figure 2, the architecture implementing μDTN.

However, while μDTN reduces to an absolute minimum the protocol stack, the code is still too large to be deployed on the sensors of IOT-Lab. We have therefore chosen to offer a lighter version of μDTN, we name nanoDTN.

We synthesize in table 2 results analysis of existing implementations compared to metrics that we are interested in for our deployment.

3.2.5 nanoDTN

We named our architecture, nanoDTN, since it is based on μDTN architecture, which we removed features while retaining basic interoperability provided by the μDTN. This implementation being too large to be carried on WSN430, we created ours, lighter, for deployment. We kept the operating system Contiki [17] using the C language and therefore allows the use of multiple standard libraries.

Table 2: Comparison of different implementations for deployment on WSN430

Implantation	Language Appropriate	Memory pre-allocated	Size code	Bundle Protocol interoperable	Layer low convergence
ION [9]	✓	✓	X	✓	X
DTN2 [16]	X	X	X	✓	X
IBR-DTN [11]	✓	✓	X	✓	X
μDTN [13]	✓	✓	X	✓	✓
DTNLite [14]	✓	✓	✓	X	✓
ContikiDTN [15]	✓	✓	✓	✓	X

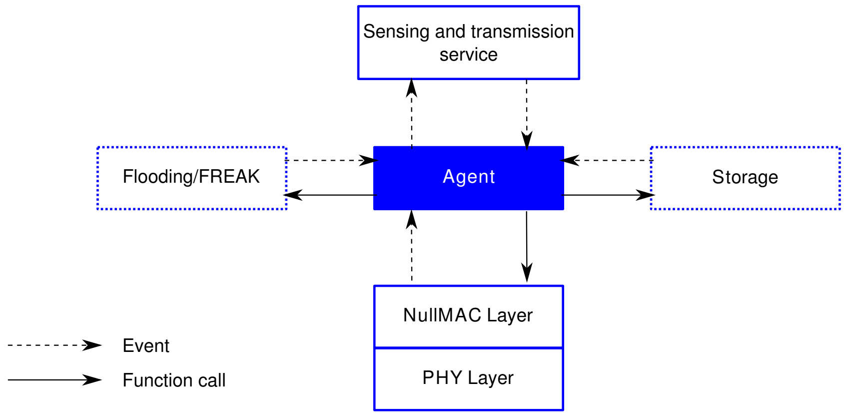

Figure 3: nanoDTN architecture

We started by removing the functions of reports. These features are useful for custody. From there, we have also chosen to remove the management of custody that could no longer exist without the reports. We note that this is not because we delete the functionality of custody that nodes can no more carry Bundles from other sources. The difference is that now a node carrying a bundle will no longer be able to withdraw it from its memory after custody taken for this Bundle by another node. A Bundle will be removed because delivered to the destination, acknowledgment reception for this Bundle or expiration of its lifetime.

In order to reduce the memory footprint, we have also chosen to delete the redundancy function that allowed not to convey a new bundle that could have been delivered. We follow on functionalities removal by deleting the Neighbor Discovery. So we chose periodic transmissions.

Our field of application does not require reliability. We decided to use only one MAC layer, which does not support carrier sense and provides no reliability. This MAC transmits data to the upper level and the lower level. We have also considered that these sensors could not have other features than collection and transmission of data. So we have restricted to a service composed of sensing, gathering and periodic transmissions. Finally, we restricted the routing to the routing functionality that is the simplest, a routing by flooding. This is a simple routing, robust communications intensive but very few resources, so adapted to our context.

Figure 3 shows the architecture of the nanoDTN implementation.

We have never changed functionality inherent to the Bundles headers to maintain interoperability with other implementations of the Bundle Protocol.

We will modify the routing function to test the FREAK scheme that we define next.

4 Using meetings with gateway = FREAK

In [18], the authors outline an assumption by writing that *the future rate nodes meet can be roughly predicted by the previous rate.* We consider this hypothesis and suggest a mechanism we call FREAK.

4.1 The FREAK algorithm

FREAK (Frequency Routing, Encounters And keenness) is so named because it offers a frequency-based routing meeting and is optimistic. Our proposal is optimistic because unlike solutions like Encounter-Based Routing (EBR) [18], MaxProp [19] or Probabilistic Routing Protocol using History of Encounters and Transitivity (PRoPHET) [20] that use information meeting on the set of nodes of the network, our algorithm uses only the frequency of meeting with all destination nodes. This restriction allows to respect the constraint on the memory of nodes we have analysed earlier.

One might expect that replication is not appropriate in a context where memory limitations are strong. However, by depriving themselves of this mechanism, we decrease the number of contacts per message. With the replication, a message has as many opportunities as a carrier node has encounters. In the case of transmissions without replication, there is only one message bearer. The transmission reduces the load but it also reduces the number of useful contacts per message and provides worse than using replication mechanisms, more specifically in an opportunistic environment.

Our idea is to assume that some terminals will encounter more often the destination than others. Rather than determining the number of meetings with each node [18], we calculate the average frequency of encounters with the gateway. The best relays are the nodes meeting more often destination. In conclusion, the higher the frequency of meeting with the destination, the better this relay is. The FREAK operation is simple. When two nodes meet the node which sees less often the base station sends copies of its Bundles to another. The algorithm 1 summarizes this operation.

The metric is updated only when the node meets the gateway and carries Bundles to deliver to it.

4.2 The communication protocol for FREAK

FREAK is the algorithm which, based on a metric of encounters of nodes with the destination, decides whether a node has to transmit Bundles or not. We need to use a protocol to exchange this metric when two nodes are in contact.

We define our discovery protocol such as each node is periodically sending a control message containing its address and the value of its metric. When a node receives such a message, it creates a new message containing its own value of the metric and sends this answer to the former node.

Depending on the values of the metric, the FREAK algorithm computes on each node whether it has to send or receives Bundles. There is no reliability provided, but the goal of such a scheme is to decrease the number of useless transmissions when using a routing scheme such as flooding and also to use low memory and computation skills to fit on a sensor.

Algorithm 1 The FREAK mechanism

Let A be the local node
$nbrContacts = 0$
$freq = 0$
for all met node (called B) **do**
  if B is the destination **then**
    $nbrContacts + +$ # *If this is the destination the metric is updated*
    $freq = \frac{nbrContacts}{CurrentTime}$
    Send all Bundles # *and all Bundles are transmitted to B*
    Delete delivered Bundles
  else
    if $contact_{freq}(A) < contact_{freq}(B)$ **then** # *If met node sees the destination more frequently*
      send Bundles to B # *then all Bundles are transmitted*
    else
      wait Bundles from B # *otherwise, expecting to receive those from B*
    end if
  end if
end for

Table 3: table summarizing the characteristics of the scenario

Traffic renewal	Periodical
Lifetime	Few hours
Sensor Types	Static and mobile
Volume of data	10 bytes

5 Selected scenario

We select a scenario of tests compliant with an urban environment. We consider that a portion of the nodes would remain static and that other nodes which could be carried by citizens, vehicles, public transports would be mobile. Then, the mobile nodes would serve as gateways or as Mobile Ubiquitous LAN Extension (mule) to gather data from static nodes. Data collected by such a system might represent a huge volume, but from a lot of sources providing low volume information, such as measurements of temperature, noise or atmospheric pollution.

5.1 Traffic

Measurements are performed periodically. The extent of the monitored areas is large. It will be necessary that the network be able to handle data from different sources.

We assume that the lifetime of the collected data should be within a few hours. While gathering data from an urban environment, a part of the collected data might represent an interest for near real time applications while most of the collected data is transmitted, then stored to be analysed offline. Applications such as smart parking or sharing transport systems require near real time data.

We synthesize the choices made in the table 3 .

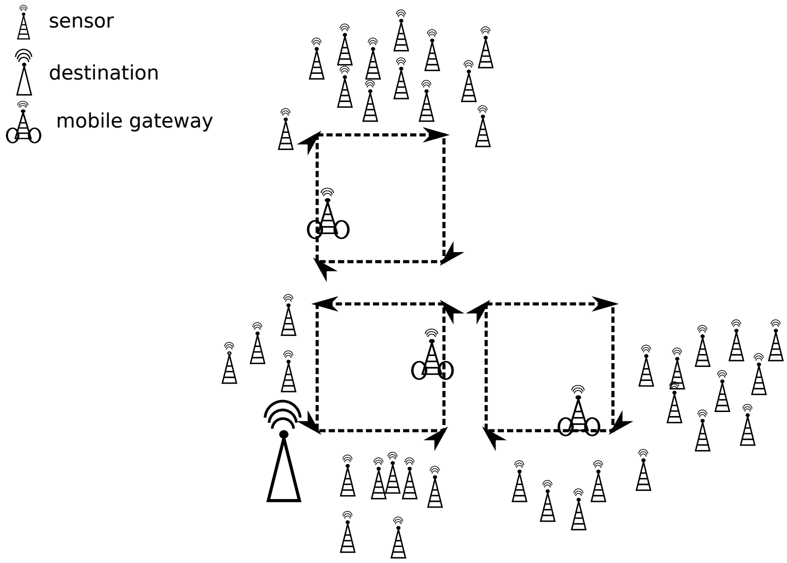

Figure 4: Topology Scenario

5.2 Hardware constraints

In the network of sensors, nodes are static or mobile. In the case where the sensors are mobile, this would mean that they would disposed on human beings or vehicles. Having no knowledge of the frequency of meetings of these elements, we cannot program adequate awakenings. In order to reduce the number of missed contacts with other terminals, we choose to let radio equipment on continuously.

5.3 Topology

As we mentionned earlier, the network will deal with static and mobile nodes. In order to take benefit from DTN and the Bundle Protocol, some parts of the network will remain disconnected. Only some mobile nodes will handle a partial connectivity between two parts of the network.

The topology we thought about to test our mechanism contains three network segments with static nodes. These nodes can only communicate with nodes from another segment thanks to one of the mobile nodes. We also consider three mobile nodes which will have a pattern whose parameters such as speed and period will change from one test to another.

This topology is shown on figure 4.

We note that each mobile node visits only one static area. We will be able to analyse the performance of a network depending on the use of a basic routing scheme such as flooding or on the use of a mechanism which tries to infer the future based on measurements of the past encounters.

5.4 Technological choices

The sensors that we will use will be the WSN430 type. These sensors are the equipments of the IOT-Lab platform.

Table 4: Selected settings

Sensors memory	48 kB
Number of sensors	12
Mobile nodes	3
Static nodes	9

These devices have limited memory capacity, with 48 kilobytes of ROM and 10 kilobytes of RAM. Depending on the version of the WSN430, transmissions are either made with a frequency of 868 MHz or 2400 MHz with a 802.15.4-compliant chipset [21]. These motes are the IOT-Lab nodes.

5.5 Performance to evaluate

We define here the performance that we will evaluate with our tests.

- The delivery ratio of Bundles which reached the destination.
- The overhead of communications is the ratio of transmitted Bundles within the network over the number of delivered Bundles.
- The delivery delay between the source and the destination.

In the next section, we develop how the tests were run and analyse the results of the experiments.

6 Experiments and results analysis

The deployment of mobile nodes in IOT-Lab was not yet realised. We made the choice to test on a Contiki emulator, Cooja [17], the implementations of the algorithm and protocol by identifying the most constraining scenario of the defined scenario.

6.1 Experiments

Cooja is a java-based platform which allows to emulate the behaviour of several sensor nodes. We have to select the type of node that we use to run the test, then we compile the code as if a true hardware sensor were used. In order to handle the mobility, we had to add a plugin to this platform and we created one scenario with a deterministic mobility. This mobility is a periodic pattern corresponding to the one defined in the figure 4.

We figured out that for each disconnected network, a middle sink would be used. The purpose of this middle sink is to concentrate data from this part of the network and to relay to the mobile node. We use Bundles with a size of 80 Bytes at the middle sink node. Then this node receives data from 7 static nodes and keeps it in memory waiting for the mobile gateway to receive data. The middle sink node is also named the source node, because from the point of view of the mobile part of the network, it is this node which sends traffic into that part.

We use 3 mobile nodes. The first one, named A, is periodically in contact with the source node, the second and the third ones. The second one, named B, is a mobile node which meets the sink node and the first mobile node. The third node, named C, meets the first and second nodes. Such a node does not help the network to reach the sink, but other nodes might transmit

Table 5: Network performance

	Flooding	FREAK
Delivery ratio	98.55%	89.50%
Overhead	3.03	3.25

Bundles to it. Our goal is to analyse how the two selected routing schemes perform deal with such a situation.

When we ran the first validation tests, we concluded that we needed to make two alterations to the selected routing schemes. If a node relays a Bundle to another node and keeps it in memory – as most DTN schemes do – custody and/or report schemes should be managed in the network, otherwise the source nodes expect an acknowledgement from the final destination and new Bundles are lost. We have already analysed that the memory limitations of the sensor nodes could not allow to use one of these schemes. We made the choice to remove a Bundle from memory when a node transmits a Bundle to another node.

The second alteration that we made is on the FREAK algorithm itself. If a relay node does not generate data for the destination, then its frequency value will remain the initial one. If this node meets the destination and has no Bundle for the sink, then it will not be able to identify this node as the destination and will not update its metric. We altered the FREAK algorithm to allow a node to transmit data to a neighbour having the same frequency value. This is a main alteration of the FREAK algorithm. Indeed, in its original behaviour, only nodes meeting the destination more frequently collect data from other nodes. With this second version of FREAK, only nodes having worse statistics of encountering the destination will not collect data.

These first tests of the implementations revealed that when a network is composed of heterogeneous nodes in terms of mobility and traffic generation, some mechanisms might require an alteration or the network would end up in a deadlock. We also realised that the concessions made on the knowledge of the network because of memory limitations can impair the performance of the schemes used within a network. For example, we can not keep a frequency value of each met node as well as we can not use acknowledgements, while we already analysed that using acknowledgements within a DTN increases performance of the network [22].

Now that the routing schemes have been altered to allow the network to send data between source and destination, we analyse the performance of the network with the two routing schemes.

6.2 Results analysis

The selected topology allowed us to select two different mobility scenarios. Either A meets C, then A meets B or A is in contact with B before C. In the former situation, the node A transmits all its Bundles to this node with the two routing schemes and no Bundles reach the destination. The table 5 and the figure 5 present the results of the latter scenario on the mobile part of the network.

This result can be explained because the flooding protocol sends Bundles to all met nodes and since Bundles are not kept in memory after transmission, because of the deadlock that occurs otherwise, only the first node gets data. When FREAK is used, Bundles are transmitted between A and C because the nodes are set up with the same initial values. Then with the alterations made on FREAK, node A decides to transmit Bundles to C and has no more Bundle to send to the sink node.

From the table 5, we see that the performance of Flooding are better than the ones of FREAK. Indeed, more Bundles are delivered and less transmissions per Bundle are used to

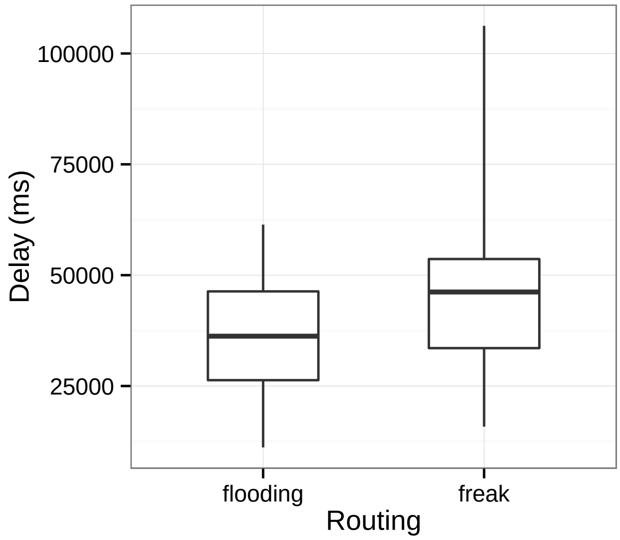

Figure 5: Delivery delays statistics with the routing scheme used

reach this result. Another fact that we can point out is that the overhead is very close to 3 which is the lower bound in our mobile network. Indeed the nodes A, B and the destination have to receive a Bundle to complete a transmission and this generates an overhead of 3.

When we look at the figure 5, we see that deliverys occur faster with Flooding than with FREAK. The boxplot represents the minimum, first quartile, median, third quartile and maximum for the delivery delay. Most delays are within the same range with the two routing schemes but there is an advantage to Flooding while the mobility scenarios are exactly the same. One reason that could explain this difference is the handshake realised by FREAK. Both protocols used beacons to detect neighbours, but with FREAK, when a node receives a beacon, it has to answer with a beacon so that both nodes which one has to receive data. This handshake reduces the duration of each contact where Bundles can be transmitted. Then some Bundles might miss an opportunity because of the delay introduced by the handshake of FREAK.

After the analysis of the three metrics, we decided to search into the logs, the reason of the difference in delivery ratio and overhead. We noticed that no Bundles were transmitted to the node C and there were a little bit more Bundles in the buffers when FREAK was used at the end of the scenario. Nevertheless, this difference is not enough to explain the difference in delivery ratio. We realised that since some Bundles could not be transmitted because of efficient contacts lasting less when using FREAK, from time to time the buffers might have exceeded their capacity and when new Bundles are transmitted to a node with a full buffer, these Bundles are lost.

To improve the performance of the network, we think that the discovery DTN protocol should use beacons containing a field indicating the amount of free memory for the Bundles buffer. Then any DTN routing protocol should use this information to decide whether it shall transmit or wait for another contact to transmit data. The very low memory capacity of the sensors is the main constraint to DTN applications for IOT when mobile nodes are involved. Based on the results of this study, we consider that mobile nodes implementing a DTN protocol stack should have better computation capabilities and larger memory than the basic static sensor nodes.

7 Conclusion

We defined in this report, an architecture compliant with several constraints of IOT purpose, such as mobility management and low memory devices, and fitting several applications it is not designed for one particular application. The implementation of such an architecture was analysed and lead to the definition of a new one, nanoDTN. Based on this architecture, we propose a DTN routing algorithm to determine whether or not a node shall send its Bundles during a contact.

The experimentations that we lead, allowed us to determine the key elements to improve IOT with DTN. The mobile nodes should have higher computation skills and larger memories that sensing nodes. Indeed, using DTN involves a big use of the memory to implement the algorithms and protocols. Furthermore, mobile nodes have to collect, store and relay data from several places. The goal of a static sensor is to last for a long period of time. For the mobile gateways, their purpose is to allow to transmit efficiently data. We also figured out that within a mobile DTN network implying sensors, the use of custody or acknowledgements would improve the network performance; and this can be achieved only with nodes having enough memory to store the information needed by these mechanisms. A scenario of IOT with heterogeneous nodes in terms of mobility might imply heterogeneous characteristics for the nodes, and then DTN mechanisms would benefit to scenario with static and mobile sensors.

References

[1] B. Guo, D. Zhang, Z. Wang, Z. Yu, and X. Zhou, "Opportunistic iot: Exploring the harmonious interaction between human and the internet of things," *Journal of Network and Computer Applications*, vol. 36, no. 6, pp. 1531 – 1539, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804513000052

[2] M. Younus, A. Minhas, M. Javed, and A. Naseer, "Eear: Efficient energy aware routing in wireless sensor networks," in *7th International Conference on ICT and Knowledge Engineering*, 2009, pp. 57–62.

[3] R. Haider, M. Javed, and N. Khattak, "Eagr: Energy aware greedy routing in sensor networks," in *Future Generation Communication and Networking (FGCN)*, vol. 2, 2007, pp. 344 349–344 349.

[4] R. Kacimi, R. Dhaou, and A.-L. Beylot, "Load balancing techniques for lifetime maximizing in wireless sensor networks," *Ad Hoc Networks*, vol. 11, no. 8, pp. 2172 – 2186, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570870513000930

[5] O. Incel, A. Ghosh, B. Krishnamachari, and K. Chintalapudi, "Fast data collection in tree-based wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 11, no. 1, pp. 86–99, 2012.

[6] D. P. Dallas and L. W. Hanlen, "Optimal transmission range and node degree for multi-hop routing in wireless sensor networks," in *Proceedings of the 4th ACM Workshop on Performance Monitoring and Measurement of Heterogeneous Wireless and Wired Networks*, ser. PM2HW2N '09. New York, NY, USA: ACM, 2009, pp. 167–174. [Online]. Available: http://doi.acm.org/10.1145/1641913.1641937

[7] B. Elbhiri, S. Fkihi, R. Saadane, N. Lasaad, A. Jorio, and D. Aboutajdine, "A new spectral classification for robust clustering in wireless sensor networks," in *6th Joint IFIP Wireless and Mobile Networking Conference (WMNC)*, 2013, pp. 1–10.

[8] N. Kushalnagar, G. Montenegro, and C. Schumacher, "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals," RFC 4919 (Informational), Internet Engineering Task Force, 2007. [Online]. Available: http://www.ietf.org/rfc/rfc4919.txt

[9] S. Burleigh, in *4th IEEE, Consumer Communications and Networking Conference, 2007. CCNC 2007.*

[10] K. Scott and S. Burleigh, "Bundle Protocol Specification," RFC 5050 (Experimental), Internet Engineering Task Force, 2007. [Online]. Available: http://www.ietf.org/rfc/rfc5050.txt

[11] S. Schildt, J. Morgenroth, W.-B. Pöttner, and L. Wolf, "Ibr-dtn: A lightweight, modular and highly portable bundle protocol implementation," *Electronic Communications of the EASST*, vol. 37, pp. 1–11, January 2011. [Online]. Available: http://journal.ub.tu-berlin.de/eceasst/article/view/512

[12] W.-B. Pöttner, J. Morgenroth, S. Schildt, and L. C. Wolf, "An empirical performance comparison of DTN bundle protocol implementations," in *ACM MobiCom Workshop on Challenged Networks (CHANTS)*, Las Vegas, Nevada, USA, 2011. [Online]. Available: http://www.ibr.cs.tu-bs.de/bib/acmdl/?http://dl.acm.org/authorize?6580950

[13] G. von Zengen, F. Büsching, W.-B. Pöttner, and L. Wolf, "An Overview of μDTN: Unifying DTNs and WSNs," in *Proceedings of the 11th GI/ITG KuVS Fachgespräch Drahtlose Sensornetze (FGSN)*, Darmstadt, Germany, September 2012. [Online]. Available: http://www.ibr.cs.tu-bs.de/papers/buesching-fgsn2012.pdf

[14] R. Patra and S. Nedevschi, "Dtnlite: A reliable data transfer architecture for sensor networks," in *8 th International Conference on Intelligent Engineering Systems*, 2004.

[15] M. Loubser, "Delay tolerant networking for sensor networks," Master's thesis, Swedish Institute of Computer Science, January 2006.

[16] "Dtn2," April 2014. [Online]. Available: http://www.dtnrg.org/wiki/Dtn2Documentation

[17] "Contiki," December 2015. [Online]. Available: http://www.contiki-os.org/index.html

[18] S. Nelson, M. Bakht, and R. Kravets, "Encounter-based routing in dtns," in *Proceedings IEEE INFOCOM*, 2009, pp. 846–854.

[19] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "Maxprop: Routing for vehicle-based disruption-tolerant networks," in *Proceedings IEEE INFOCOM*, 2006.

[20] A. Lindgren, A. Doria, E. Davies, and S. Grasic, "Probabilistic Routing Protocol for Intermittently Connected Networks," RFC 6693 (Experimental), Internet Engineering Task Force, 2012. [Online]. Available: http://www.ietf.org/rfc/rfc6693.txt

[21] "Wsn430 hardware in iot-lab," July 2015. [Online]. Available: https://github.com/iot-lab/iot-lab/wiki/Hardware_Wsn430-node

[22] P. Raveneau, R. Dhaou, E. Chaput, and A. Beylot, "Dtns back: Dtns broadcasting ACK," in *IEEE Global Communications Conference, GLOBECOM 2014, Austin, TX, USA, December 8-12, 2014*, 2014, pp. 2789–2794. [Online]. Available: http://dx.doi.org/10.1109/GLOCOM.2014.7037230

RESEARCH CENTRE
GRENOBLE – RHÔNE-ALPES

Inovallée
655 avenue de l'Europe Montbonnot
38334 Saint Ismier Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-0803