A multi-level FREAK DTN: Taking care of disconnected nodes in the IoT

Conference Paper · November 2016

DOI: 10.1109/NOF.2016.7810111

Patrice Raveneau
Université de La Rochelle

Emmanuel Chaput
Institut de Recherche en Informatique de Toulo…

Riadh Dhaou
Ecole Nationale Supérieure d’Electrotechnique…

A multi-level FREAK DTN: Taking care of disconnected nodes in the IoT

Patrice Raveneau
Univ La Rochelle – L3i Lab – EA 2118
F-17000 La Rochelle, France
patrice.raveneau@univ-lr.fr

Emmanuel Chaput , Riadh Dhaou and André-Luc Beylot
Université de Toulouse – IRIT – ENSEEIHT
Toulouse, France
Email: {chaput, dhaou, beylot}@enseeiht.fr

***Abstract*—Crowdsensing is, for a few years, a hot topic. Until now, research on crowdsensing mainly focused on scenarios with devices such as smartphones with huge memory and high computive skills. With the development of the Internet of Things (IoT), crowdsensing can be envisaged with other constraints. Indeed, some IoT nodes are mobile but with limitations about storage and processing capabilities, then connectivity disruptions might occur between the nodes. These issues are tackled by a Disruption Tolerant Networking architecture. In this article, we focus on a subset of IoT, Mobile Sensing Networks (MSN). We propose then, a mechanism which respects the constraints of the nodes and maintains high performance. This mechanism, the multi-level FREAK, uses the mean frequency of contacts with the destination. The metrics drives the transmission. Since some nodes might not meet the destination nor nodes meeting the destination, we had the idea of a multi-level metrics to allow these "disconnected" nodes to transmit data to the destination. We evaluate our proposal through simulations based on several real mobility traces. Our solution outperforms reference replication and quota-based DTN solutions.**

***Index Terms*—Internet of Things, Disruption Tolerant Networking, Opportunistic network, Data transmission.**

I. Introduction

The interest for the Internet of Things (IoT) has been growing in the last few years. Cisco forecasts that the IoT will generate $19 trillion over the next decade [1]. The spectrum of applications which can benefit from IoT is very large. Indeed, these applications go from personal private uses to urban and smart cities purposes [2].

With IoT for a personal usage, people can follow metrics such as how many steps they have accomplished along the day or if their potted plants require more water. In a smart city context, the applications might use static sensing nodes for example with smart metering or air pollution monitoring [3].

Nevertheless, when talking about static nodes IoT, it is compulsory to analyse before deployment the limits of the network to sense this data in order to optimize the placement of nodes. Mobile IoT does not suffer from this constraint. Considering an air pollution monitoring scenario, it is possible to collect data measurements with mobile sensors which would be installed on city bus or on bikes of a bicycle-sharing system. A system combining sensors from buses and from bikes would allow to take measurements from several patterns with the buses and measurements from other locations thanks to those on the bikes.

The devices used for such an application have to be cheap since their number would be high. They also have to be energy-efficient; then, it is compulsory to use low energy consuming communications protocols. This constraint implies that a cellular or Wi-Fi network could not be used because consuming too much energy. Protocols such as the IEEE 802.15.4 fit for this situation. Having recourse to this family of protocols implies that the network would suffer from many link disruptions. This is why we focus here on a Disruption Tolerant Network (DTN) for IoT with opportunistic contacts.

With an IoT-DTN, some contacts between nodes might be somehow periodic, while using buses to carry the sensing nodes; but other contacts are opportunistic. Then, as often when dealing with DTN, the aim is to be able to benefit from useful contacts to reach the destinations.

As the authors of [4] explain, DTN routing protocols can be classified into three categories which are based on forwarding, replication with or without quota. Forwarding-based routing protocols only keep one copy of each message within the network unlike replica-based protocols where the number of copies might increase.

In [5], the authors propose a forwarding-based DTN routing protocol named Link Contact Duration-based Routing Protocol (LCD). This solution uses the intercontact duration and the number of messages to determine which messages have to be sent to which relay. LCD achieves good performance but it is compared only to other forwarding-based protocols with one mobility dataset. Maybe some replications-based protocols could achieve better performance. Furthermore, the mobility dataset used for simulations is from a bus network which is predictable in advance with the timetables. That is why, we use several mobility datasets, so that our conclusions do not apply only for one kind of mobility.

Quota and flooding based protocols are the two sub-categories of replica-based routing protocols. MaxProp [6] and Prophet [7] are the most known flooding-based routing protocols while Spray And Wait [8] is a very efficient and simple quota-based routing protocols with a first phase where the replicas are spread among the nodes then a second phase where each replica might only be sent to the destination. Epidemic [9] is also a flooding-based protocol where at each contact nodes send to each other every single message that the other node does not have in memory and never carried. Encounter-Based Routing (EBR) [4] is a quota-based routing protocol which is very elegant by taking into consideration the past encounters with destinations while not wasting memory resource with unnecessary replicas.

MaxProp, Prophet and EBR provide high performance in a mobile and opportunistic networks but they are very complex. Indeed, MaxProp uses an ordered queue based on the estimated likelihood of a path to the destination and Prophet keeps track of each probability of encounter with every node. The problem with complexity is that some elegant solutions are too complex to fit on IoT devices such as WSN430 [10] which present very limited ROM (48 kB) and RAM (10 kB) [11].

Most DTN studies focus on peer-to-peer communications meaning that the data exchanged in the network goes from random sources to random destinations. These settings allow to model communications to exchange data between entities in the network such as text messages between two humans or pieces of a navigation map within a vehicular network. Nevertheless, when considering crowdsensing, the main goal is to be able to retrieve data from the sensing nodes to a gathering machine whose aim is to store the data. In the specific case of crowdsensing with IoT, the energy constraint limits the ability of nodes to communicate with an infrastructure network. We consider that one or few nodes of the mobile sensing network serves as gateway between mobile IoT nodes and an infrastructure network able to communicate with the storing server.

This paper is organised as follows. We present in a first time the proposed mechanism in section II then we describe the tools and datasets we used to evaluate our proposal in section III before analysing the results of the simulations in section IV and finally we conclude on what we learned from this study in section V.

II. Where the multi-level FREAK comes from

One of the most important limitation of the context of this study is the low memory and computation skills of the devices intended to be used in mobile sensing IoT. Because of this hardware constraint most smart DTN solutions cannot be implemented on the nodes. We use for comparison some of the best mainstream DTN routing protocols but we have to keep in mind that the algorithms used by these solutions are too greedy with resources. The comparison allows us to check if our proposal performs well compared to solutions requiring a lot more resource to work.

As the authors of [4] claim, it might be possible to infer future encounters based on the ones that occured in the past. This assumption is even more true when dealing with human-centered activities. Indeed, the life of any human presents patterns which are repeated through time with small and bigger periods. From this angle, a solution like the FREAK one [12] appears to be a good option. Nevertheless, we can wonder if by increasing a little bit the complexity the performance could not become greater.

In FREAK, the main idea is that only the frequency of encounters with the set of destinations (which are considered as aggregated from the routing point of view) matters. But, even if this is an important parameter, we have to take into consideration the fact that nodes might have a mobility pattern which leaves them too far from the destination; and even too far from nodes close to the destination. A simple example would be a scenario where two nodes N and M never meet the destination D and where a third node L would meet the destination. N never meets L, but M does. With the FREAK solution, N shall not send to M because its value of the metric is bad as the one of N. But since M can forward data to L which meets the destination, N should transmit data to M to reach the destination.

That is why, we inspired from the results on κ-vicinity from [13] to extend the simple FREAK solution [12] in a mechanism which would take benefit from multi-hop contacts. With this solution, that we named multi-level FREAK DTN, we should be able to decrease the problems, like the example we described earlier, that might occur in some networks for a certain period of time.

Now that we explained the motivations to propose our multi-level FREAK solution, we describe the behaviour of the protocol. Before the transmission of any data packet (named Bundles when using the DTN Bundle Protocol [14]), when two nodes meet, they exchange their values of the metrics. The metrics are in the same spirit of the one used by the simple FREAK version. We use an array of frequencies with groups. The first value of the array is the frequency of encounters between the node and the set of destinations. The second value of the array is the frequency of encounters between the node and a set of nodes meeting the destination; and so on. We limited to an array of eight values in order to limit the impact on the memory of the nodes. Furthermore, we believe that given a network not exceeding the size of a city, eight hops is enough to cover the whole area, even with cheap mobile devices within an infrastructureless network.

Once the nodes exchanged their vector of metrics, they run the algorithm 1 to update their vector and make a decision on whether sending replicas of messages they carry or not. The vector contains the identifier of the node and it is used to determine if the destination is met or not.

After the presentation of the motivations for our proposal and how the multi-level scheme works, we present the evaluation environment.

III. Evaluation environment

Our evaluation is realised through simulations with the DTN simulator The ONE [15]. This simulator does not take into account problems from the MAC layer assuming that any problem of this level would be tackled by a protocol from this level. Our goal is to compare the multi-level version of FREAK with the simple one and to analyse if its performance is close to

Algorithm 1 Multi-level FREAK Transmission Decision

Let **A** be the local node
$i = 0$
while $(i < 8)$ **do**
  $nbrContacts[i] = 0$
  $freq[i] = 0$
end while
for all met node (named **B**) **do**
  if B is the destination **then**
    $nbrContacts[0] + +$
    $freq[0] = \frac{nbrContacts[0]}{CurrentTime}$
    Send all Bundles
    Remove all delivered Bundles
  else
    $i = 1$
    while $(i < 8)$**and**$(freq[i](\mathbf{A}) == 0)$**and**$(freq[i-1](\mathbf{B}) == 0)$ **do**
      $i + +$
    end while
    if $(i < 8)$ **then**
      $nbrContacts[i] + +$
      $freq[i] = \frac{nbrContacts[i]}{CurrentTime}$
    end if
    $i = 0$
    while $(i < 8)$**and**$(freq[i](\mathbf{A}) > freq[i](\mathbf{B}))$ **do**
      $i + +$
    end while
    if $(i < 8)$ **then**
      send Bundles to **B**
    else
      wait for **B** to send Bundles
    end if
  end if
end for

performance from more complex and unimplementable DTN routing protocols.

We use several mobility traces for our simulations in order to limit the importance of the mobility patterns of one trace in a specific context. We present first these datasets and analyse their differences and similarities. Thanks to this analysis we can check if some mobility characteristics might have an influence on network performance.

A. Datasets presentation

We gathered several mobility traces and contact traces from the CRAWDAD website then used the Adyton tool [16] to generate contact traces for each trace that we selected. The Adyton tool provides a DTN simulator we found less powerful in terms of settings compared to The ONE. Then we modified our contact traces in a format compatible with The ONE.

The traces that we used were collected at conferences such as Infocom05 (Inf5), Infocom06 (Inf6), Sigcomm09 (Sig9). This type of traces are very specific because the public is attending an event. We also used traces coming from campuses such as the MIT Reality dataset (MIT) and Milano (Milano). Finally, we used the Cambridge (Camb) and DieselNet (DN) dataset which were collected at a city level for the former and thanks to a DTN installed on buses of UMass Amherst campus for the latter.

We realise an analysis on the number of contacts that each node makes during the whole simulations. We will not compare the maximumn and minimum values of each dataset

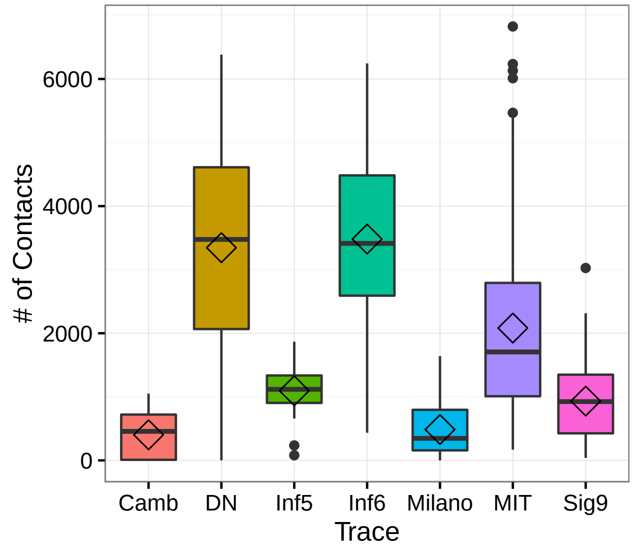

Fig. 1. Comparison of distributions of number of Contacts per mobility dataset

for the number of contacts since as we can see on Table I the durations of simulations are not in the same range. We analyse the trends of the distributions thanks to boxplots presented in Figure 1.

From this figure, we note that the shorter a scenario is and the closer are the quantiles of the number of contacts per node. Indeed, the four traces with the smallest durations (Camb, Inf5, Inf6 and Sig9) are the ones with the tightest boxplots while the three others which last a lot more in time present loose boxplots. This trend is not impacted by the number of nodes. This can be explained by the fact that on small periods people will have very similar mobility patterns (going to work/conference, getting home, etc.); while once the duration of the mobility trace is longer than a few weeks the mobility patterns start to diverge. Then for the three other traces, nodes do not move in the same manner and this can explain why the whiskers are bigger.

TABLE I
IMPORTANT VALUES OF MOBILITY TRACES

Traces	mean(s) intercontact	max(s) intercontact	# nodes	Duration (s)
Camb [17]	1,846	588,020	54	987,529
DN [18]	2,659	879,7164	37	10,635,474
Inf5 [17]	213	182,542	41	254,150
Inf6 [17]	102	65,337	98	337,418
Milano [19]	2,385	84,0503	49	1,632,979
MIT [20]	9,444	5,673,540	97	24,428,517
Sig9 [18]	182	158,400	76	320,774

The mean, represented with diamonds, is most of the time close to the median meaning that the distribution of number of contacts per node is uniform. For the datasets Milano and MIT where the mean is between the median and the third quartile, there are more nodes having a lot of contacts than nodes having very few contacts.

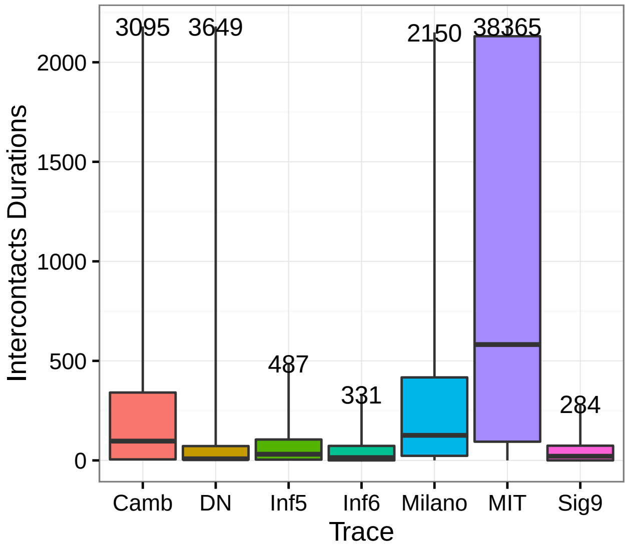

Fig. 2. Comparison of distributions of inter-contacts durations per mobility dataset

We analysed the number of contacts per node since contacts are a key element for opportunistic networks. We also analyse on Figure 2 the distribution of inter-contacts duration per node for each dataset.

On this figure we indicated the maximum value of the inter-contact duration at the end of the top whisker because some values were too high to get a clear interpretation.

Here, we see that there is no strong correlation between the distributions of number of contacts and inter-contacts durations. Indeed, we note a strong correlation between the three conferences datasets with low inter-contacts durations, that we can explain because of the small covered area. The distribution of inter-contacts of the bus network has very low quartiles with maximum and mean values far upper from the third quartile, as are the distributions for Camb, Milano and MIT.

From this analysis, we think that the performance results of our simulations run on conferences datasets might be very specific because the mobility and contacts look like they are very different to the other mobility traces. The four other traces which do not share an identical scenario have nonetheless similar statistics on the metrics we analysed.

We now present the choices made for our simulations.

B. Simulations parameters

We focus on a scenario of IoT crowdsensing with cheap devices. We assume that the measurements are made periodically every hour. We do not take into account applicative mechanisms that would decide to not transmit a measurement because the value is the same as the previous one. We make this assumption because we want to know the performance of a network with a given load. If some mechanisms allow to decrease the load, then the network should perform even better.

TABLE II
SIMULATIONS SETTINGS

Messages size (Bytes)	50
Nodes buffer	20 messages
TTL (hours)	12 – 60
Message generation period per node (hour)	1

We consider one scenario with one destination and we select three different destinations based on the analysis of the number of contacts of each node. The first run is when the node having the highest number of contacts is selected as destination. The second run is with a destination selected within the nodes having the smallest number of contacts. And the third run is with the destination randomly picked among nodes having a number of contacts close to the first quartile. By varying the type of destinations, we want to analyse if it is better to have a node with a lot of encounters as a destination or as a relay.

We summarize the simulations settings in Table II. We consider that for applications whose aim is to monitor the environment, a periodic measurement with a period of one hour allows to get an accurate information on the measured metrics. Since this data is not critical, the delay to collect it is not limiting. Then we choose time to live (TTL) values high and we vary the TTL between 12 and 60 hours.

We compare the multi-level FREAK mechanism to DirectDelivery (DD) where nodes transmit only to the destination, EBR, Epidemic, the simple FREAK scheme, MaxProp, Prophet and SprayAndWait (SAndW).

IV. RESULTS INTERPRETATION

Now we analyse the results of the simulations we run. First of all, we will not use figures for the results when the destination is either a node with the least number of contacts or with a number of contacts close to the first quartile of the dataset. Indeed, for these settings, the delivery ratio was around one or two percents for every routing protocol on each dataset.

From these results, we can conclude that it is far better for a network when the destination is a node making a lot of contacts. We could think that a relay node making a lot of encounters would allow the network to have high performance; but because of the limited storage of the nodes, the performance drops when the destination does not belong to the nodes making the greatest number of encounters.

After this first analysis, we focus on the delivery ratio when the TTL is 12 hours on Figure 3.

The first fact that we note on this figure is that the multi-level FREAK scheme provides always better performance than the simple version. This indicates that with the simple version, there are contacts between nodes which are not used. This is now tackled by the multi-level version. We also note that the multi-level FREAK provides delivery ratio very close to mainstream DTN routing protocols such as MaxProp and Prophet unless for the conferences scenario. We identified in the analysis on the datasets that the conferences traces had

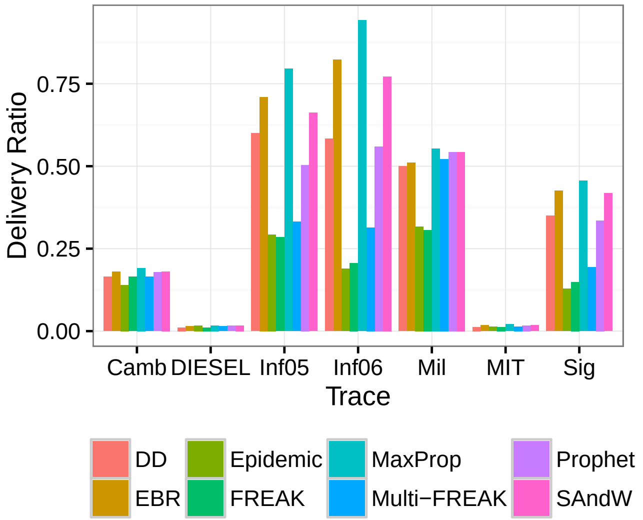

Fig. 3. Delivery Ratio with TTL = 12h

a specific mobility pattern, and this type of mobility is not convenient for low computation solutions.

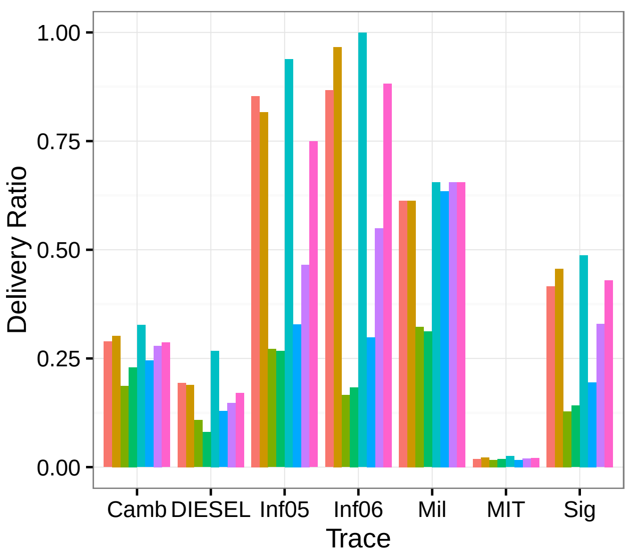

Fig. 4. Delivery Ratio with TTL = 60h

This trend is the same when we increase the TTL of the Bundles to 60 hours. The multi-level FREAK (ML-FREAK) is always better than the simple FREAK, but when it is used at a conference, the performance is not at the same level as other mainstream DTNrouting protocols. By considering the four other mobility scenarios, the multi-level FREAK provides delivery ratio in the same range of solutions like MaxProp of Prophet while our proposal is far less complex than these two solutions. On the trace coming from Milano, our solution is even better than EBR which is also more complex than ML-FREAK.

With Figures 5 and 6, we note one of the drawback of the dropping policy we selected. When a node receives data from a neighbour node and its memory is full, then it drops the messages that it had in memory to be able to get the new messages. This tends to increase the overhead which is defined as the number of exchanged messages over the number of delivered messages. Since the ML-FREAK does not order the messages to send in a smart way as MaxProp does, then the delivered messages have a higher probability of coming from nodes which are further (in terms of hops) from the destination. This tends to increase the delay of the delivered messages.

Except with the conferences traces which are maybe biased, ML-FREAK outperforms other DTN routing protocols since it provides a delivery ratio equal or greater than the one from the mainstream DTN routing protocols at a lower computation cost.

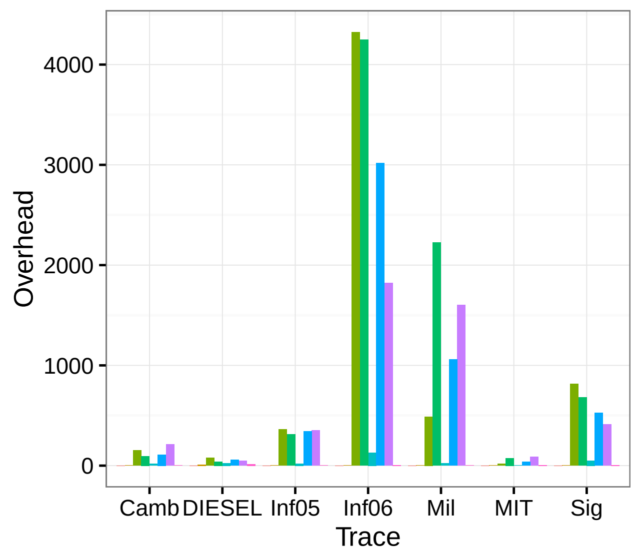

Fig. 5. Overhead ratio with TTL = 60h

V. Conclusion

To conclude this study, we proposed the multi-level FREAK protocol which can fit on small devices with low memory and computation skills and provides a performance level as high as the one of far more complex routing solutions. We also analysed some characteristics on the contacts of several reference mobility datasets and we could determine why some protocols have better performance than others when the mobility scenario is very specific.

As a perspective, we are currently implementing ML-FREAK on real devices (WSN430) and we will test the performance in real conditions. We also envisage to focus on a dropping policy and on an algorithm which would allow to improve the fairness among the sources of messages. As always, when talking about this type of devices, such algorithms should not be too complex.

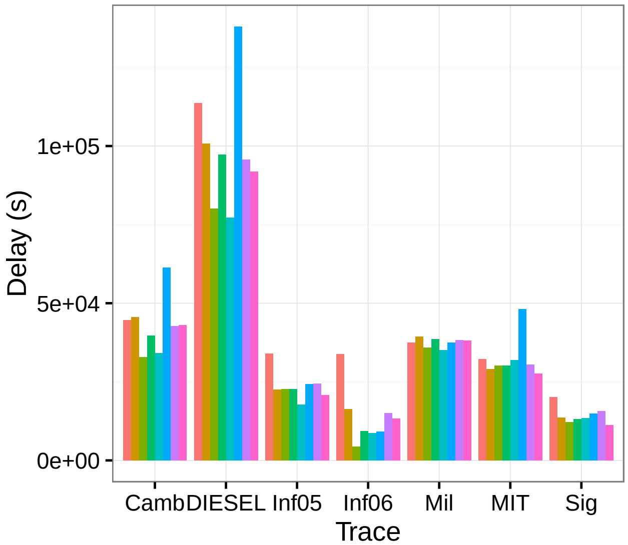

Fig. 6. Mean delay with TTL = 60h

References

[1] Cisco, "The internet of everything is the new economy." [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/enterprise/cisco-on-cisco/Cisco_IT_Trends_IoE_Is_the_New_Economy.html

[2] C. Gomez and J. Paradells, "Urban automation networks: current and emerging solutions for sensed data collection and actuation in smart cities," *Sensors*, vol. 15, no. 9, pp. 22 874–22 898, Sep 2015.

[3] A. Boubrima, F. Matigot, W. Bechkit, H. Rivano, and A. Ruas, "Optimal deployment of wireless sensor networks for air pollution monitoring," in *2015 24th International Conference on Computer Communication and Networks (ICCCN)*, Aug 2015, pp. 1–7.

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

[5] K.-H. Jung, W.-S. Lim, J.-P. Jeong, and Y.-J. Suh, "A link contact duration-based routing protocol in delay-tolerant networks," *Wireless Networks*, vol. 19, no. 6, pp. 1299–1316, 2013. [Online]. Available: http://dx.doi.org/10.1007/s11276-012-0534-0

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

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

[8] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: An efficient routing scheme for intermittently connected mobile networks," in *Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-tolerant Networking*, ser. WDTN '05. New York, NY, USA: ACM, 2005, pp. 252–259. [Online]. Available: http://doi.acm.org/10.1145/1080139.1080143

[9] A. Vahdat and D. Becker, "Epidemic routing for partially-connected ad hoc networks," Tech. Rep., 2000.

[10] FIT/IoT-LAB, "Hardware wsn430 node." [Online]. Available: https://github.com/iot-lab/iot-lab/wiki/Hardware_Wsn430-node

[11] P. Raveneau and H. Rivano, "Experiments and results on DTN for IOT III Urbanet collaboration," 472, Research Report RT-0472, Dec. 2015. [Online]. Available: https://hal.inria.fr/hal-01257644

[12] P. Raveneau, E. Chaput, R. Dhaou, and A. L. Beylot, "Freak dtn: Frequency routing, encounters and keenness for dtn," in *2014 IFIP Wireless Days (WD)*, Nov 2014, pp. 1–3.

[13] A. Tatar, T. Phe-Neau, M. D. de Amorim, V. Conan, and S. Fdida, "Beyond contact predictions in mobile opportunistic networks," in *Wireless On-demand Network Systems and Services (WONS), 2014 11th Annual Conference on*, April 2014, pp. 65–72.

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

[15] A. Keränen, J. Ott, and T. Kärkkäinen, "The ONE Simulator for DTN Protocol Evaluation," in *SIMUTools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques*. New York, NY, USA: ICST, 2009.

[16] N. Papanikos, D.-G. Akestoridis, and E. Papapetrou, "CRAWDAD toolset tools/simulate/uoi/adyton (v. 2016-04-21)," Downloaded from http://crawdad.org/tools/simulate/uoi/adyton/20160421, Apr. 2016.

[17] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, "CRAWDAD dataset cambridge/haggle (v. 2009-05-29)," Downloaded from http://crawdad.org/cambridge/haggle/20090529, May 2009.

[18] A.-K. Pietilainen and C. Diot, "CRAWDAD dataset thlab/sigcomm2009 (v. 2012-07-15)," Downloaded from http://crawdad.org/thlab/sigcomm2009/20120715, Jul. 2012.

[19] P. Meroni, S. Gaito, E. Pagani, and G. P. Rossi, "CRAWDAD dataset unimi/pmtr (v. 2008-12-01)," Downloaded from http://crawdad.org/unimi/pmtr/20081201, Dec. 2008.

[20] N. Eagle and A. S. Pentland, "CRAWDAD dataset mit/reality (v. 2005-07-01)," Downloaded from http://crawdad.org/mit/reality/20050701, Jul. 2005.