DTNs BACK: DTNs Broadcasting ACK

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

***Abstract*—The mobile context of Mobile Wireless Sensor Networks (MWSN) limits the existence of a direct route from source to destination. A Disruption Tolerant Networking (DTN) architecture fits the requirements for such a context where messages need to be stored, carried and forwarded. For this kind of DTN applications, the goal is to achieve a high delivery ratio at low transmission cost with the lowest latency. Some DTN routing protocols use this ACK information to decrease the number of useless transmissions. Nevertheless in memory-constrained environments, the proportion of memory allocated to ACKs is a problem to study. This paper focuses on the use of acknowledgements (ACKs). We model the network with a Markov chain, to study the effect of ACKs on buffer time occupancy. Finally, an extensive set of simulations is run to analyse the influence of the memory proportion allocated to ACKs, on the performance.**

***Keywords*—Wireless sensor networks, Disruption Tolerant Networking, Modeling**

I. INTRODUCTION

Static and Mobile Wireless Sensor Networks (MWSNs) have a large applications spectrum. These applications are either population monitoring, hazard control, healthcare monitoring, exploration or military operations. Furthermore, a Navigant Research analysis released in May 2013 indicates that smart cities total revenue from 2013 to 2020 will surpass $117.3 billion [1]. The wide applications spectrum and the forecast of the revenue growth from one field of applications indicate that this topic shall be investigated.

MWSNs belong to the opportunistic networks category. A node cannot be sure to find another one, neither after a fixed delay nor in a specific location. Traditional routing protocols fail at computing a path between a source and a destination because of the lack of connectivity of the network. Then MWSNs also belong to Delay/Disruption Tolerant Networks (DTNs). The existing DTN solutions should be analysed to determine their compliance with MWSNs requirements.

The context we focus on is a monitoring system such as the population monitoring systems described in [2], [3] or vehicle-based systems such as the one in SVATS [4]. In SVATS, one of the limitations is the fact that, in a sparse context, the protection cannot be maintained. Nevertheless, such a system should be able to indicate that a stolen vehicle is moving while it should remain still. Then when a node belonging to the network receives this alert message, it can relay it to the base station. It is not compulsory to have an extra road-side network. The monitoring network is able to receive messages sent by the slave sensors. Then the vehicle can still be detected as stolen with no supplementary cost. Managing such a scenario is relevant for DTNs.

We need to focus on the solutions which allow the network to achieve high performance in this context. As stated in [5], which introduces Encounter Based Routing (EBR), DTN routing protocols can be classified into three groups: forwarding-based, replica-based and quota-based.

Forwarding-based protocols are the most used in traditional networks because the connectivity of the network is high. Such networks do not require to replicate messages to reach high delivery ratio. Replication increases the network load but not necessarily the performance, even though with a limited number of messages. Nevertheless, these protocols might miss useful contacts in a DTN scenario and are not much investigated because of this drawback. In [6], an efficient single-copy protocol is proposed. Nevertheless, the delivery ratio is not analysed.

Most DTN routing protocols are replica-based. The difference between replica-based and quota-based is that the number of replicas does not depend on the number of nodes. Epidemic [7] is the simplest replica-based protocol. Other well-known replica-based routing protocols such as Prophet [8] use the probabilities of previous encounters or MaxProp [9], which reorders the messages to get the best performance.

DTN mostly focus on quota-based protocols. The reason is that the protocol is able to estimate a quasi-optimal number of replicas. Then, the overhead decreases and the network is not congested with useless packets which would be relayed for nothing. So in a quota-based network, source nodes assign a number of replicas to each message and these replicas are spread through the network and kept when only one remains. Famous examples are Spray And Wait [10] consisting of two phases and EBR [5] which relies on an encounter value of each node with all other nodes. The authors assume that if a node meets a lot of other nodes then its likelihood of reaching the destination as a relay is high.

In [11], a quota-based protocol relying on messages Time To Live (TTL) is proposed. Messages are better scheduled and the buffer utilisation is more efficient.

The main constraint on MWSNs is the memory limitation. Indeed, equipments are small and do not possess a large memory. They cannot afford neither to stock data for a long period nor use protocols requiring considerable computation skills. Then we have to propose simple and efficient schemes to take advantage of the limited capacity of the network. A scheme which is simple to implement and does not occupy a lot of memory is the use of ACKs. In a DTN context, the concept of acknowledgement is different from the traditional one.

Indeed, the idea is not only to inform the source node that its message reached the destination. The ACKs will also help to

978-1-4799-3512-3/14/$31.00 ©2014 IEEE

decrease memory occupancy within the network. Every carrier of a message can free memory as soon as it receives an ACK for this message. However, to be useful, acknowledgements (ACKs) have to be kept in memory such as an ACKs carrier will be able to forward the delivery information to each potential message carrier. The drawback is that some memory will be needed to store the ACKs. Nonetheless, ACKs are far smaller than messages. The advantages and drawbacks of this scheme have to be analysed. This scheme is neglected in previous works. Some protocols may use acknowledgements, but the impact of ACKs is not investigated as it deserves.

This paper is organised as follows. First, we present the proposed scheme and why it has been analysed in section II. A definition of the problem is made in section III. Analytical results are discussed. Finally, an extensive set of simulations results are analysed, in section IV, to get a more precise idea of the real behaviour of the scheme.

II. Broadcasting ACKs back in the network

The use of ACKs is not new for DTNs, but it is not commonly considered in the related work. Some protocols such as MaxProp are able to use ACKs. Unfortunately, the literature does not present neither how they are managed when the memory is almost full nor their impact on the network performance.

We propose to analyse the influence of ACKs on several protocols. The aim of this paper is to propose a protocol and a scheme which could fit on a small sensor node. We will focus on simple as well as on complex protocols, in order to analyse if the benefits of acknowledgements work for each protocol.

Whatever the protocol to which we decide to apply ACKs, each time a node carrying ACKs meets another one, it transmits all ACKs that the second node does not possess. This means that we choose to exchange ACKs in an epidemic manner. ACKs are generated by the destination each time a message reaches the destination. Even though the final node receives an already received message, it has to send an ACK for this message. However, the destination does not keep track of the already acknowledged messages. This means that when a node meets the destination, it does not receive ACKs of all already delivered messages.

This paper tackles two main questions: whether there is a need to add ACKs in a memory-constrained scenario and how the load is impacted by a dedicated memory management based on ACKs. If the gain by using ACKs is too low, adding memory to nodes will be mandatory.

Beside this ACKs analysis, we study the performance of a simple protocol whose aim is to be implemented on small sensors that we proposed earlier [12]. This protocol uses the same principle as in [5] which is that the future can be predicted from the past. Nevertheless we do not use the same metric. Indeed, within a MWSN all nodes do not play the same role. Then, we only focus on the encounter rate with the base stations. This protocol is named FREAK because it relies on Frequency Routing based on the Encounters and is Keen. Indeed, it is keen to assume that future depends on past events, but it is also real. In real life, people are more often in the neighbourhood of home or work rather than in a location where they go only on holidays.

The proposed contribution is replica-based. Nonetheless, the defined metric allows to decrease the number of transmissions. Indeed, nodes copy and forward data only to nodes with a better metric than theirs. This decreases the number of replicas but might also decrease the delivery ratio. We will compare the proposition to mainstream DTN protocols. We consider this comparison in a memory-constrained environment where heavy protocols could not be implemented. We hope that the performance of the proposition is close to DTN standards one.

III. Modelling the network

The assumptions made in this section are maintained for the remainder of the paper unless we specify so.

We consider N mobile nodes with unlimited buffer capacity. These nodes are referred to as relays, sources or carriers. We consider one static node which is the sink or destination of the network. The buffer size is the same for each node in the network unless the sink node which has an infinite buffer. The inter-contacts between nodes are considered exponentially distributed with a mean frequency λ.

We focus on the latency and the sojourn time of messages within the network. We consider two DTN routing protocols to analyse their performance:

- Direct Delivery: the sources transmit only to the final destination.
- Epidemic [7]: at each contact nodes forward messages that the other node does not possess.

As a first analysis, the buffer capacity of mobile nodes, is considered infinite. Then the performance of one message does not depend on the queue occupancy. We assume contact durations long enough to exchange required data. Hence, we model the number of replicas of a message in the network thanks to a Markov chain.

A. Classic version: no ACKs

The Markov chain for the Direct Delivery case is obvious. It contains only two states. The original one when a message is generated at a source and the absorbent final state when the message is delivered at the destination. The transition probability is of course the mean contact rate between nodes λ. In this context, the mean sojourn time is equal to the mean latency.

For the Epidemic analysis, the Markov chain is a bit more complex, as we can see on Figure 1. Each state represents the number of nodes carrying a message and the fact that at least one replica has reached the destination. In the classic version of this protocol, ACKs are not used. Then, a node may carry a message, deliver it and carry it again because it does not know that the message was carried before. Hence, the number of states in the chain is equal to twice the number of mobile nodes. Then, we have several cycles within the chain. When we look for the latency of a message, the half of the chain in Figure 1 is composed of absorbent states. Absorbent states are those of the bottom line. Nevertheless, when looking at the mean sojourn time of messages, there is only one absorbent state, corresponding to the state where there is no more message within the network except at the

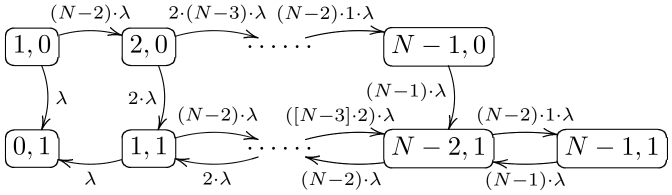

Figure 1. Epidemic Markov chain for delivery delay

destination. Then, it is possible to simplify the Markov chain by removing most absorbent states and aggregating them with other states as shown on Figure 2.

Concerning the mean delivery delay, all states of the bottom row on Figure 1 are absorbent. Indeed, we focus on the time for one replica to reach the destination. Each absorbent state represents that the destination has been reached by a replica. Then, once a packet arrives at the destination, the network evolution is no more useful.

We normalise the transient rates (the sum exiting a state equals one). Then, the probability to reach one absorbent state $[k, 1]$ is easily derived:

$$P\left[k,1\right] = \frac{1}{N-1} \tag{1}$$

Consequently, we derive the mean delay to reach one absorbent state is given by:

$$D[k,1] = \sum_{i=1}^{k+1} \frac{1}{i \times \lambda \times (N-i)} \tag{2}$$

Hence, the mean delivery delay is:

$$D_d[N] = \frac{\sum_{k=0}^{N-2} P[k,1] \times D[k,1]}{\sum_{k=0}^{N-2} P[k,1]} = \frac{1}{N-1} \times \sum_{k=0}^{N-2} D[k,1] \tag{3}$$

Expression 3 can be simplified by following the path rather than computing first, the probability of the path. Then we get:

$$\begin{aligned} D_d[N] = & \frac{1}{(N-1)\times\lambda} + \frac{(N-2)}{(N-1)} \times \frac{1}{2\times(N-2)\times\lambda} \\ & + \frac{(N-2)}{(N-1)} \times \frac{(N-3)}{(N-2)} \times \frac{1}{3\times(N-3)\times\lambda} \\ & + \ldots + \frac{1}{(N-1)\times\lambda} \end{aligned}$$

The previous expression is simplified in Equation 4:

$$D_d[N] = \frac{1}{(N-1)\times\lambda} \times \sum_{k=1}^{N-1} \frac{1}{k} \tag{4}$$

Giving the expression for the mean delivery delay, we focus on the mean sojourn time in the network. We want to estimate the time required to remove a message from the network. In other words, the mean time for each replica to reach the destination. Then, we only have one absorbent state (it models that there is no more message in the network). The Markov chain on Figure 2, represents the same system as on Figure 1 except we do not differentiate with or without delivery. In order to better understand how to derive the mean time for a replica to reach the destination, the first chain is proposed.

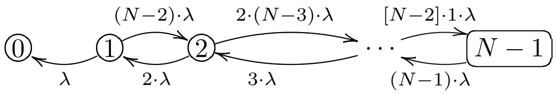

Figure 2. Epidemic Markov chain for mean sojourn time

We want to calculate the mean sojourn time of a message in the network. To derive this value, we compute the mean time to reach the state 0 of the second Markov chain. Since, the Markov chain is composed of cycles it is possible to have infinite length paths between the initial state and the absorbent state. Since the chain in Figure 2 is finite and includes one absorbent state, the mean time to go from the initial state to the absorbent one converges.

To compute the mean sojourn time, it is necessary to find the mean time to go from the initial state to the absorbent one. To do so, we note $\overline{t_i}$ the mean time to reach the state 0 from the state i. By analysing the chain in Figure 2, we get the following set of equations.

$$\begin{cases} \overline{t_1} & = & \frac{1}{(N-1)\times\lambda} + \frac{N-2}{N-1} \times \overline{t_2} \\ \overline{t_2} & = & \frac{1}{2\times(N-2)\times\lambda} + \frac{1}{N-2} \times \overline{t_1} + \frac{N-3}{N-2} \times \overline{t_3} \\ & \vdots & \\ \overline{t_{N-1}} & = & \frac{1}{(N-1)\times\lambda} + \overline{t_{N-2}} \end{cases} \tag{5}$$

This system is linear and we solve it for a set of values of network sizes to compare the classic protocol and its BACK version.

Figure 4 presents the evolution of the delay and the sojourn time for a mean inter-contact rate equals to 0.1 while the number of nodes grows. The results of this figure will be analysed later with the results of the BACK mechanism.

B. BACK version

The aim of this paper, in the following, is to evaluate the influence of using acknowledgements on the two metrics we focused on earlier. In this section, we assume that each time a message is delivered, the destination sends an ACK which is broadcasted within the network. We represent the Markov chain on Figure 3, to calculate the mean delivery delay and sojourn time when using ACKs.

The states of the chain represent the number of replicas of a message and the number of acknowledgements of this message. These two values are bounded by $N-1$. We note, on Figure 3, that there is no cycle. Then we have a finite chain with a finite number of paths, then all times are bounded. We point out that when a Markov chain has a finite number of states with absorbent states and a path between any transient state and at least one absorbent state, we can also conclude that all times are bounded. The transitions are only possible to a state possessing one more message copy, one more ACK

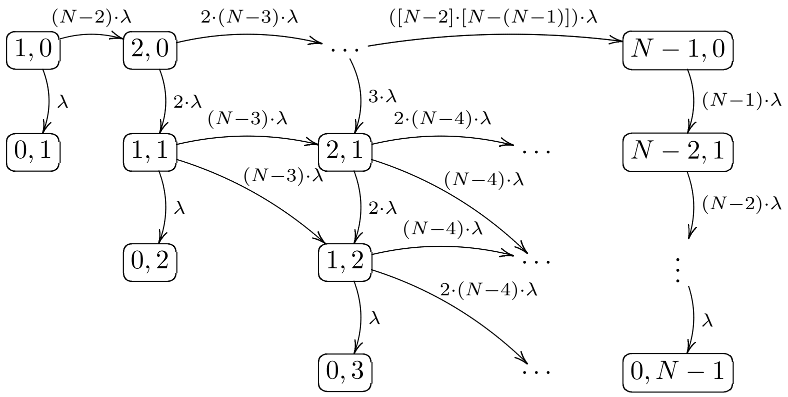

Figure 3. Epidemic Markov chain with BACK scheme for delivery delay and sojourn time

or one less message and one more ACK. Nonetheless, when a state does not possess any ACK, it is not possible to go to a state with a supplementary ACK without decreasing the number of message copies. We need to reach the destination at least once to get one ACK.

The mean delay is the sum of delays to reach a state with one ACK since this means that the destination was reached once. We note that the formula for the delay is the same with and without ACKs. It is logical that ACKs do not improve nor worsen the delay.

The mean sojourn time is the sum of the delays of each path beginning in the initial state and arriving in a state with zero message copy weighted by the probability to follow this path. This time is computed by exhaustively listing each path.

Figure 4 represents the evolution of the delay and sojourn times for the Epidemic with and without broadcasting ACKs. The mean delay of the direct delivery protocol appears also on this figure. We can notice that, when the number of nodes grows, the mean delay decreases with the Epidemic, while the Direct Delivery proposes the worst achievable delay.

However, when looking at the curves of the mean sojourn time, we notice that, when using ACKs, the sojourn time increases, and then decreases while the network size grows. This is explained because when the size of the network grows, the number of paths increases, but the number of nodes carrying messages does not evolve as fast because of the low probability to broadcast ACKs. Indeed, the number of nodes carrying ACKs are on average low compared to the number of nodes carrying messages. After, the size of the network keeps growing, and more and more nodes will be able to broadcast ACKs, then the mean sojourn time decreases.

Concerning the curve of the sojourn time without ACKs, we observe a sharper increase than with ACKs. We conclude from this analysis that ACKs are able to decrease the sojourn time. This will help for a finite buffer scenario, as messages will remain shorter in memory on nodes.

We analysed the influence of ACKs in an infinite buffer scenario with inter-contact between nodes distributed exponentially. We now need to check the influence of ACKs in a more realistic scenario through simulations.

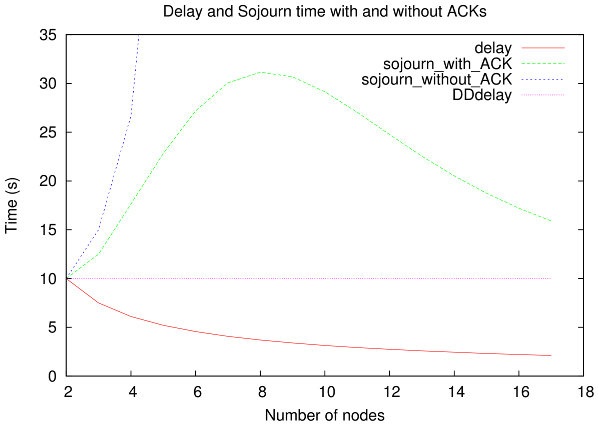

Figure 4. Delivery and Sojourn Delays for Epidemic with and without ACKs for $\lambda = 0.1$

Table I. Simulations Parameters

Number of nodes	100
Mobility Models	GaussMarkov
Simulation Duration	1 day
Buffers size	16, 40, 80, 160 Bundles
Buffer proportion for ACKs	[0 - 90%]

IV. Simulations

A. Environment

We use the ONE simulator [13] to run simulations. Table I summarises the parameters of the simulations.

In the simulations, $N-1$ mobile sensor nodes and one static base station are considered. Mobile nodes have buffer limitations. The volume ratio between an acknowledgement and a message is a tenth. Then, for each non-carried message, a node is able to store up to ten ACKs.

Buffer sizes are very small. This choice is motivated by an implementation we performed of a lightweight version of the Bundle Protocol on a micaZ mote that the remaining available memory could manage only few tens of Bundles to few Bundles depending on the size of the Bundles. For example, when the Bundles are only one hundred bytes long, a micaZ mote is able to store less than twenty Bundles.

The proportion of memory allocated to ACKs goes from 0 to 90% in order to be able to compare the protocols with and without acknowledgements. We reach almost 100% to see how the network reacts when the data memory decreases and the ACK data increases accordingly.

We now analyse the results of the simulations.

B. Results analysis

The metrics, we focus on, are the delivery ratio, the delay, the overhead ratio and the sojourn time. We hope that the use of ACKs is going to improve delivery ratio, overhead ratio and sojourn time without worsening the delay. We focus on the metrics when no ACKs are used by increasing the memory to analyse its impact on network performance.

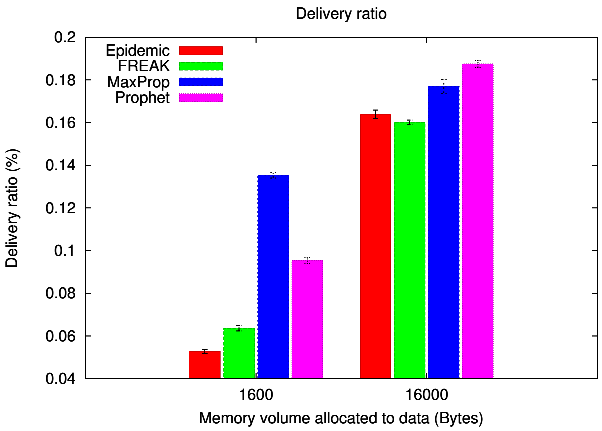

Figure 5. Delivery Ratio with 1.6 KB and 16KB in memory without ACK

Figure 5 presents that, while the capacity is increased in a ratio of ten, the delivery ratio is at most multiplied by three and at worst increased by 33%.

For the remainder of the results, ACKs are used. We see in Figure 6, that the delivery ratio with 1.6 KB is far much greater with ACKs than with 16KB and no ACKs. In Figure 6, the delivery ratio of FREAK is close to flooding-like protocols such as Epidemic, and protocols based on analysis of past encounters such as MaxProp or Prophet. It is shown in Figure 8 that with FREAK, when the memory dedicated to ACKs is low, the resulting overhead is light and lower than with other protocols.

Nonetheless, the improvement on delivery and overhead is done by increasing the sojourn time and the delivery delay as we notice on Figures 7 and 9. This indicates that FREAK does not select the shortest path in terms of delay but intends to select paths that decrease the number of transmissions. With the number of vehicled ACKs increasing, the delivery ratio decreases. That is the reason of the delay decrease. FREAK delivers less messages because of lack of memory; then delivered messages are the ones with shortest delays. Finally the proposed scheme is able to achieve a delivery ratio in the same range as complex protocols while decreasing the number of transmissions. Furthermore, the complexity of

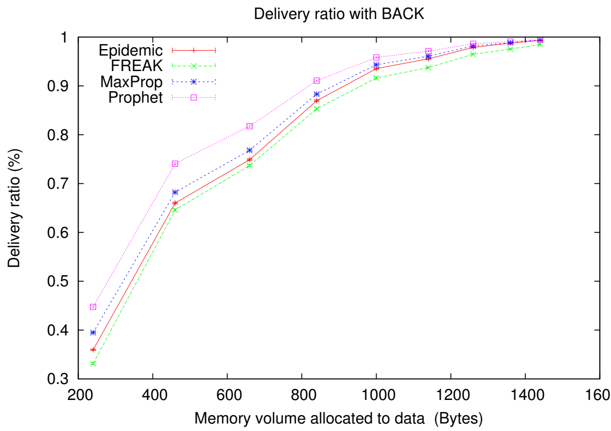

Figure 6. Delivery Ratio with 1600 Bytes maximum in memory with ACK

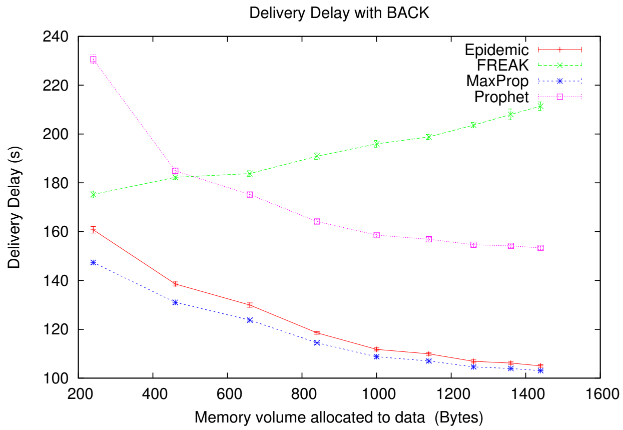

Figure 7. Delivery Delay with 1600 Bytes maximum in memory with ACK

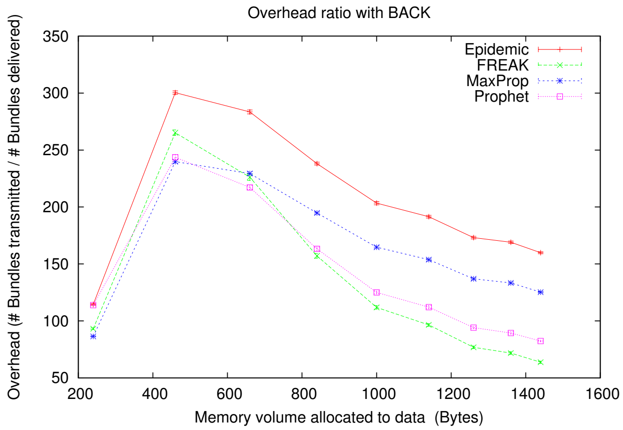

Figure 8. Overhead Ratio with 1600 Bytes maximum in memory with ACK

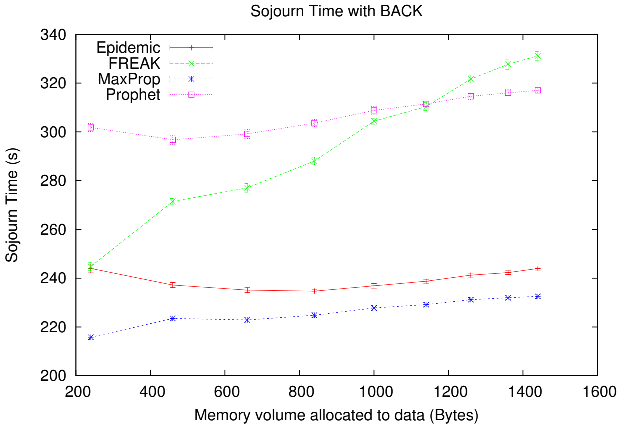

Figure 9. Sojourn Time with 1600 Bytes maximum in memory with ACK

FREAK is very low compared to other protocols and the required information to store is also very low, which perfectly suits the studied scenario of small sensors with low memory and few computation skills.

The important point for the large buffer sizes, is that the delivery ratio, when there is no ACK, is very low. There are two ranges of delivery ratio, the one without ACK, which is less than 20% and the one with ACKs close to 100%. Then, it appears that it is better to provide schemes to efficiently use the memory rather than increasing the memory. Furthermore, a memory efficiently managed might even be decreased a bit, to let some space to ACKs without degrading performance.

A complex scheduling on ACKs should be investigated in order to analyse if there is a gain. It should also be analysed if the gain is worth the complexity increase. We made the choice to use a FIFO policy related to the ACKs loss. Which is important for an ACK, is that it had been transmitted to a maximum of useless messages carriers. Then first arrived ACKs, are more likely than the ones arrived last to have been forwarded more often. Nonetheless, a simple policy based on the number of transmissions of each ACK or a loss priority depending on the arrival timestamp of the first message replica at the destination are solutions which should be investigated.

V. Conclusion

In this paper, we prove that using ACKs within DTN is a smart option, even for a memory-constrained environment, such as MWSNs. Thanks to the small size of ACKs, it is possible to reach very high delivery ratio even with small portion of the buffer allocated to ACKs. Moreover, the use of ACKs reduces the number of useless messages; inducing a decrease of overhead and, in the same time, an increase of the delivery ratio. Furthermore, the FREAK proposition is a simple scheme providing performance in the same range as complex protocols such as MaxProp, with a smaller overhead. Then, it is shown that the proposed protocol reaches its purpose to provide performance in the same range as mainstream DTN protocols, with less computation skills and memory required.

As a perspective of this work, we envisage to seek for other schemes to better manage network resources and specifically the memory. Indeed, in order to decrease the number of messages and ACKs transmissions, a scheme relying on the frequency of inter-contact with the destination, combined to the timestamp of the last visit at destination, deserves to be analysed. For example, nodes knowing their mean inter-contact period with the destination, could remove messages from their buffers if they remained for a period of time greater than a threshold, based on the mean inter-contact period with the destination. This discarding would not be based on the lifetime of the Bundles, but on the occupancy of nodes buffer to prevent congestion. The problem with such schemes is to remove messages which were never delivered to the destination while congestion has not yet occurred. It is also possible to investigate a combined use of ACKs and such a scheme, by decreasing the buffer space allocated to ACKs and increasing the threshold to remove messages. Another interesting solution would be to limit the number of ACK transmissions based on the mean messages propagation in the network. Another idea, more specific to monitoring scenarios, is the use of cumulative ACKs. This means that an ACK corresponding to a message from one source, would acknowledge all messages from this source generated prior to this message.

References

[1] "Smart city technologies will generate more than $117 billion in revenue between 2013 and 2020," http://www.navigantresearch.com/newsroom/smart-city-technologies-will-generate-more-than-117-billion-in-revenue-between-2013-and-2020.

[2] J. Krause, S. Krause, R. Arlinghaus, I. Psorakis, S. Roberts, and C. Rutz, "Reality mining of animal social systems," *Trends in Ecology & Evolution*, vol. 28, no. 9, pp. 541 – 551, 2013.

[3] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet," *SIGPLAN Not.*, vol. 37, no. 10, pp. 96–107, Oct. 2002.

[4] H. Song, S. Zhu, and G. Cao, "Svats: A sensor-network-based vehicle anti-theft system," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, 2008, pp. 171–175.

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

[6] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Single-copy routing in intermittently connected mobile networks," in *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on*, 2004, pp. 235–244.

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

[8] 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.

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

[10] 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.

[11] A. T. Prodhan, R. Das, H. Kabir, and G. C. Shoja, "Ttl based routing in opportunistic networks," *Journal of Network and Computer Applications*, vol. 34, no. 5, pp. 1660 – 1670, 2011.

[12] P. Raveneau, E. Chaput, R. Dhaou, and A.-L. Beylot, "FREAK DTN: frequency routing, encounters and keenness for DTN," 2013.

[13] 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.