Distributed Broadcast Scheduling in Mobile Ad Hoc Networks

Carlos Henrique Pereira Augusto and José Ferreira de Rezende
GTA/COPPE/UFRJ - Universidade Federal do Rio de Janeiro
Rio de Janeiro, Brazil
Email: {`chenrique,rezende`}`@gta.ufrj.br`

***Abstract*—Data and control messages broadcasting is a widely used mechanism in network applications and protocols, which can have latency requirements for the information delivery. One solution to achieve low latencies is to solve the Minimum-Latency Broadcast Scheduling (MLBS) problem. However, MLBS is a NP-Complete problem, thus some works propose approximation algorithms. This paper presents simulation results of well-known flood control mechanisms over IEEE 802.11 scheduling (CSMA/CA), which show that simple heuristics can provide acceptable latencies. Thus, a distributed scheduling mechanism that requires only partial topology knowledge is proposed and evaluated when combined with flood control mechanisms on CSMA/CA and TDMA networks. Results show the good performance of the proposal compared to theoretical limits of centralized algorithms.**

I. INTRODUCTION

Network flooding is a mechanism that aims to deliver messages to every node in the network. These mechanisms become essential in ad hoc networks due to their dynamic topology and lack of infrastructure. In these networks, they are widely used in routing protocols [1], [2], service discovery [3], [4], and information search and dissemination [5].

The most simple and used way to perform a flooding is called Blind flooding. However, this technique may provoke *broadcast storms* [6], in which a large number of redundant transmissions occurs. In contention-based networks, broadcast storm generates a high level of dispute and an increase in collisions, while in TDMA networks, a large number of slots would be needed to schedule all these transmissions. In both cases, the limited resources available are consumed and, hence, the latency on message delivery is increased.

The control of these harmful effects of broadcast storms can be done through two different approaches: by selecting the forwarding nodes [6], [7], [8], [9], [10], called here flood control approach, or by ordering the forwarding nodes transmissions, called here scheduling approach. Although orthogonals, the two approaches have similarities, as shown in results of Sections III and V, since the scheduling approach implicitly assume the choice of only few nodes to forward messages.

As stated above, these two approaches have different goals. The first one attempts to reduce the number of forwarded messages. The optimal solution to this problem is to seek a MLST (Maximum Leaf Spanning Tree) on the network graph having the flooding source as the root node. This solution is optimal in terms of forwarded messages since leaves do not forward messages. Thus, if the number of leaves is maximal the number of forwarded messages is minimal. However, the implementation of this solution is not practical since it is NP-hard [11]. The other approach aims to choose a transmission order amongst the forwarding nodes that leads to the Minimum-Latency Broadcast Scheduling (MLBS), which is also a NP-Complete problem [12]. An evaluation of this problem is the focus of this paper.

However, one important observation is that an appropriate choice of forwarding nodes can lead to satisfactory values for the flooding latency, even not being the target of this approach. To show this, in the first part of this work, we measure the flooding latency through simulation when different flood control mechanisms are associated with the random distributed scheduling of the IEEE 802.11 CSMA/CA.

From these results, some hypotheses are formulated, which has motivated the proposal of a new distributed scheduling mechanism based on local information with limited knowledge of the topology. This mechanism is jointly evaluated with CSMA/CA and TDMA media access networks. In the latter networks, the decision about using a slot is completely local and independent, which implies that in the case of incorrect scheduling choices, collisions and losses can occur.

The remainder of this paper is organized as follows. Section II presents related works. Section III describes the latency results obtained with proposals that only perform flood control, but not scheduling. Section IV presents concepts and algorithms of the proposed distributed scheduling mechanism. Section V shows the simulation results of our proposal compared with the theoretical values of centralized proposals. Section VI brings the conclusions, indicating the main contributions of this article, and suggests some future works.

II. RELATED WORKS

The MLBS problem is NP-Complete [12]. So, the complexity of an algorithms that can produce a solution is non-polynomial. Thus, literature presents some approaches to the problem, such as approximation algorithms, optimization methods and computational intelligence techniques, as mentioned below.

This research was supported by CAPES, CNPq, FAPERJ and FINEP.

It is interesting to note that it is easy to obtain a lower bound to the MLBS problem, which is the lowest achievable latency by an optimal algorithm in an appropriate network topology. This limit is the network radius R, i.e. the distance in number of hops of the farthest network node from the flooding source. As shown below, some algorithms have latency limits as function of R, or R and other network parameters. In addition to this lower limit, it is trivial to identify the number of nodes (N) as an upper bound, since one can always execute a correct scheduling if allocates one slot to each node.

The MLBS problem can be defined as a graph-coloring problem [13], where the solution corresponds to the smallest number of colors. If wireless networks are modeled as Unit Disk Graphs (UDGs), a protocol interference model can be used, where two nodes cannot be scheduled simultaneously if there is an edge between them, or they have at least one concurrent neighbor. Other more restrictive models can be used, for example taking into account the aggregate interference of all concurrent transmissions, also called physical or SINR-based interference model [14].

Transmission scheduling is automatically performed in CSMA/CA networks due to the carrier sense and random back-offs. Although efficient, in terms of delivery rate, CSMA/CA mechanism can impose long delays in the flooding in congested and/or dense networks. Due to this limitation, it is suitable to use a deterministic media access such TDMA for applications that require a bounded latency for flooding. However, in such cases, a centralized or distributed mechanism to perform scheduling is required. In this context, some works propose mechanisms or algorithms for this purpose by using numerous techniques.

The work in [13] deals with the MLBS problem using UDG and proposes three centralized approximation algorithms called Basic Broadcast Schedule (BBS), Enhanced Broadcast Schedule (EBS) and Pipelined Broadcast Schedule (PBS), respectively. These algorithms start from seeking a spanning tree, rooted at the flooding source, derived from the breadth-first search (BFS) in the network graph G. The BFS also determines the distance from source to all nodes and the radius R of the network. Next, a Maximal Independent Set (MIS) of G is built, induced by the increasing order of depth of nodes.

In BBS algorithm, the authors call *dominators* the nodes belonging to this MIS, and *connectors* its parents in tree. Only these nodes forward the flooding scheduled layer by layer. At layer 0, there is only the source node, and at the layer R, no node is a *connector*. Within each layer, nodes are scheduled so that no conflicts occur. Thus, the authors show that scheduling produced by this algorithm is limited to $24R - 23$ slots.

The EBS provides an improvement in the BBS algorithm by differing in how the *connectors* are selected and scheduled. In BBS, all parent nodes of *dominators* are chosen as connectors. On the other hand, EBS uses the IMC (Iterative Minimal Covering) algorithm and produces not only a smaller scheduling, but it also reduces redundant transmissions. The authors show that this algorithm produces schedules limited to $16R - 15$ slots.

Finally, the PBS algorithm computes a tree of shortest paths from source to all *dominators*. Following, it ranks the nodes according to a proposed algorithm and uses these previous assigned rankings to order them. This allows the forwards to occur in pipeline. Hence, a node of a lower layer can receive and forward a message before an upper layer node. With PBS, the scheduling is limited to $R + O(logR)$ slots. However, in the demonstration of this upper bound, one can observe that the latency is limited to $R' + O(log(R'^3))$, where R' is equal to R or $R-1$. Therefore, for the purpose of latency calculation in the following sections, we will use the limit of $R + logR^3$ (or $R + 3logR$) slots for the PBS algorithm.

The work in [12] also proposes an approximation algorithm whose latency, according to the authors, is bounded in $O(1)$ times the optimal latency. The proposed algorithm is also based on a MIS derived from the spanning tree obtained from a BFS in the graph. Despite the work to define a distributed version of the algorithm, it is based on BFS and DFS algorithms, which are costly in terms of messages. Moreover, the work in [13] indicates that can be shown that this algorithm has a latency bounded in $648 * R$.

Otherwise, work in [15] presents an algorithm that makes use of an interference vector, which allows to quickly identify if a node will interfere with another. In that work, the main objective is to build a fast algorithm to obtain a scheduling with a controllable run-time, but it does not provide an upper limit to the scheduling obtained. The paper shows results for the scheduling size when nodes are ordered by their degree and by the number of 1 and 2-hops neighbors, with small performance difference between them.

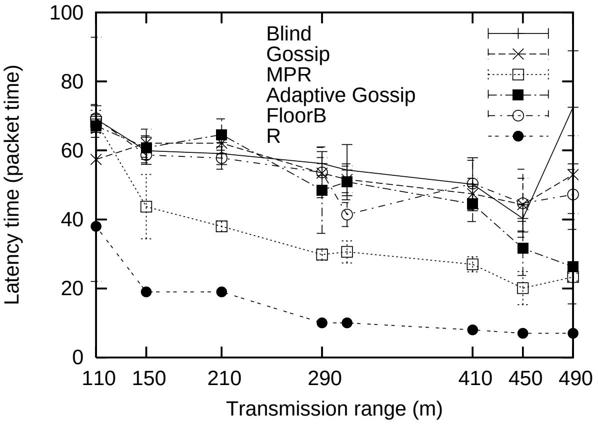
The works in [16] and [17] use other approaches to deal with the MLBS problem. In [16], fuzzy logic and neural networks tools are used, while in [17], the authors use genetic algorithms. Both works depend on the knowledge of the connectivity matrix and results are shown for only three specific networks indicated in the literature.

III. EVALUATION OF FLOOD CONTROL MECHANISMS

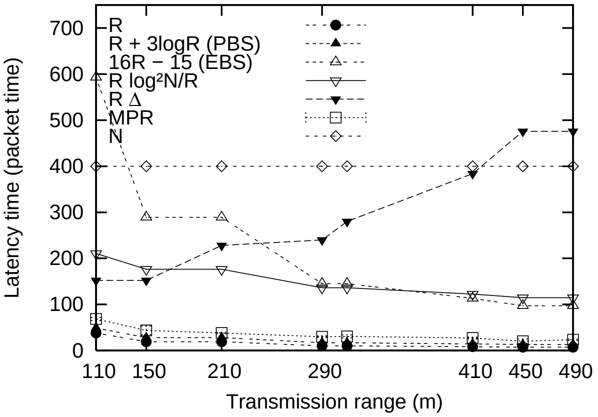
The work in [10] presents an extensive evaluation of flood control mechanisms as Blind [6], Gossip [7], Adaptive Gossip [9], FloorB [10] and MPR [8]. It evaluates the message delivery rate and saved forwarded messages of these mechanisms in several scenarios. These mechanisms make no attempt to schedule and, therefore, do not aim to reduce flooding latency. However, in this section we present a new evaluation of them using latency as the focal point. Also, we compare the results with the theoretical limits of algorithms presented in [13] and the lower bound R.

For this evaluation, these mechanisms were implemented in the ns-2 and simulated on 400 nodes placed in a 20x20 grid and spaced by $100m$. A node located in one of the corners of grid network performs flooding. In this scenario, several connectivity degrees were obtained by changing nodes transmission range from 110 to 490 meters. By using the link and physical layers of IEEE 802.11 implemented in ns-2 with their default parameterization values, except for reception

(RX) and carrier sense (CS) power threshold, we obtain the flooding latencies shown in Figure 1(a).



(a) Latency of flood control mechanisms



(b) Latency upper bound of approximation algorithms

Fig. 1. Flooding latency

We use the default 2-Ray Ground propagation model of ns-2. RX threshold was changed to vary the correspondent RX (or TX) range, and CS threshold was set to maintain the same ratio with RX threshold, thus CS range is approximately two times the RX range.

In this figure, one can observe that the Blind mechanism provides a high latency in dense networks. This occurs due to the large number of transmissions that implies a lot of contention in the medium access by IEEE 802.11 nodes. On the other hand, the MPR mechanism has a reduced latency even in dense networks, near to the lower bound (R). However, it is important to note that the MPR mechanism was designed to reduce the number of forwarded messages and not the message latency. MPR is a heuristic conceived to obtain a small number of neighbors (relay nodes) that can cover all 2-hops neighbors. This heuristic is used since finding the optimal set of relays is also a NP-Hard problem.

For comparison purposes, Figure 1(b) also presents the upper bound of some approximation algorithms, including the lower bound R and the upper bound N. One can note that limits given by these algorithms are very conservative, being closer to the upper than to the lower bounds, except for PBS algorithm [13]. Moreover, latencies provided by flood control mechanisms without scheduling are somewhat close to the lower bound, even if they were not developed for this purpose.

Therefore, one can assume that a simple heuristic to reduce the number of forwarding nodes, such as MPR, also allows a reduced latency in flood delivery. In this scenario with CSMA/CA, this can be explained by a contention decrease when a fewer number of forwarding nodes competes for the medium. Also, this reduced number of transmission nodes can be scheduled in less time but still covering all 2-hop neighbors. This process is reproduced hop-by-hop until the entire network is covered. From this observation, we made the first hypothesis to our proposed approach to the MLBS problem.

Hypothesis 1: Although orthogonal, the solutions of MLBS and MLST problems result in similar responses. Therefore, the solution of either problem is a solution, if not optimal at least close to optimal, for the other problem.

From Hypothesis 1, we developed the proposal for a simple and distributed broadcast scheduling for mobile ad hoc networks, described in Section IV.

IV. PROPOSED SCHEDULING MECHANISM

The proposed mechanism, called 2BSheld (2-hop knowledge Broadcast Scheduling with local decision), is distributed and based on the knowledge of 2-hops neighborhood from each node. This knowledge can be obtained by sending and receiving Hello messages, which contain a list of the sender's neighbors, as in the MPR, or a Bloom filter summarized list, as in FloorB [10].

Thus, we use a simple greedy algorithm, using these 1 and 2-hops neighborhood lists, where the first scheduled node will be the receiver that achieves the highest number of new (uncovered) nodes. When modeling the wireless network as a directed graph $G = (E, V)$, where V is the set of vertices corresponding to nodes, and E the set of edges representing links between nodes within the transmission range of each other, one can define as o the initial vertex of flooding, and $f_{i,s}$ the i-th vertex to forward the message of o in slot s. All i nodes that forward the message in slot s do it simultaneously. These simultaneous transmissions are possible due to spatial reuse, and it is a scheduler function to prevent simultaneous transmission of nodes that interfere with each other, and to allow those who do not interfere.

Defining as $N_1^{f_{i,s}}$ the set of 1-hop neighbors of $f_{i,s}$, a receiver $r \in N_1^{f_{i,s}}$ will be one of the j transmitters in the next slot, $f_{j,s+1}$, if $\forall v \in N_1^{f_{i,s}} \cap N_1^r$ one have $|(N_1^v - N_1^{f_{i,s}})| < |(N_1^r - N_1^{f_{i,s}})|$. Moreover, one can state that r will be $f_{k,s+n}$, where $n = |I| + 1$, and $I = \{u \in (N_1^r \cap N_1^{f_{i,s}})$ such that $|(N_1^u - N_1^{f_{i,s}})| > |(N_1^r - N_1^{f_{i,s}})|\}$.

In the case of a tie, tied nodes are scheduled in the increasing order of their network Id, i.e. if $|(N_1^v - N_1^{f_{i,s}})| = |(N_1^r - N_1^{f_{i,s}})|$, then r takes precedence over v if $Id(r) < Id(v)$.

In addition to Hypothesis 1, our heuristic is based on hypotheses 2 and 3 formulated below:

Hypothesis 2: Nodes that have the greater divergence of neighborhood to the transmitter should be at a larger physical distance from it. Therefore, nodes with great neighborhood divergence to the transmitter, and which are not neighbors, should be also at a large physical distance from each other, causing a smaller mutual interference.

Hypothesis 3: Forwarding nodes that reach a larger number of uncovered nodes also allow covering a greater percentage of network nodes in the next slot.

The basic algorithm for the 2BSheld distributed mechanism is presented in Algorithm 1.

Algorithm 1 2BSheld

```
Notation:
$f$ = forwarding node; $r$ = receiver node;
$lowPri$ = node priority in the sense of best coverage;
$lowId$ = node priority in the sense of Id;
$N_h^v$ = Set of nodes with $h$-hop distance of $v$;
$Id(x)$ = Node identifier $x$;
$next$ = number of slots to shift;
$s(x,y)$ = empirical function to compute slots
Initial values:
$lowPri = lowId = 0$
Algorithm
In each node $r$ do:
    when $r$ receives a flood message of $f$ in slot $s$, do:
        Foreach $v \in (N_1^r \cap N_1^f)$, do:
            If $|N_1^v - N_1^f| > |N_1^r - N_1^f|$
                $lowPri++$
            else
                If $|N_1^v - N_1^f| == |N_1^r - N_1^f|$ e $Id(v) < Id(r)$
                    $lowId++$
                end If
            end If
        end Foreach
    $next = s(lowPri, lowId)$
    Schedule flood forwarding in $s + next$
```

An open issue in this algorithm is the $s(x,y)$ function. The only restrictions are that it must be monotonically increasing and provide integer values for x and y integers. In order to keep the simplicity of the mechanism, the first function assessed was $s(x,y) = x + y + 1$. In Section V, other options for $s(x,y)$ will be presented and evaluated.

Figure 2 shows an example of 2Bsheld behavior. In this picture, node o launches a flooding at slot s and nodes v_1, v_2 and v_3 are first hop receivers. Node v_2 can cover three new nodes (w, y, z), node v_1 can cover two new nodes (x and y), and node v_3 can cover only one node (u), thus node v_2 is scheduled to slot $s+1$, node v_1 to slot $s+2$ and node v_3 to slot $s+3$.

V. NUMERIC RESULTS

We have implemented 2BSheld and flood control mechanisms in the ns-2 and used ten simulation runs for each input parameter. All results presented below, and including the results in Section III, are the averaged values of these runs with 95% confidence interval error bars.

The first evaluation of the 2BSheld mechanism was performed in the same grid network used in Section III. In these simulations, 2BSheld is associated with the Blind flood control mechanism, i.e. every node operates as a forwarding node, and runs on top of either CSMA/CA or TDMA media access methods. The obtained results for flooding latency (Figure 3) show that the performance of 2BSheld over CSMA/CA is worse than the one obtained by the Blind mechanism when directly applied over the same media access method.

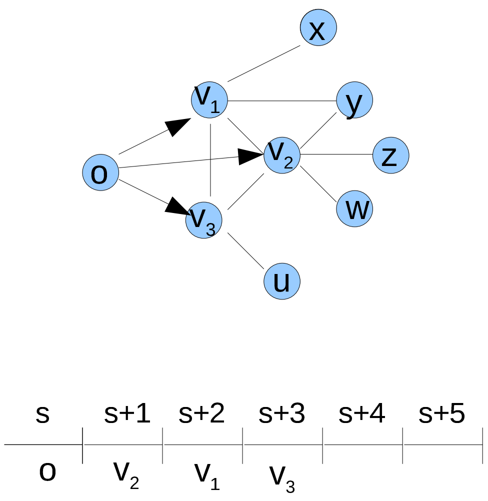


Fig. 2. 2Bsheld operation

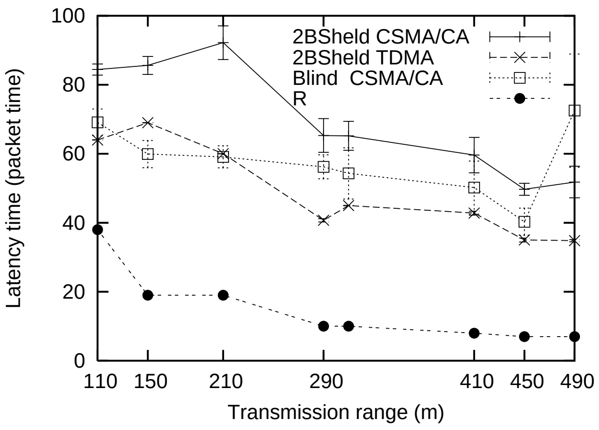


Fig. 3. Scheduler flooding latency using CSMA/CA and TDMA

This occurs because the 2BSheld scheduling mechanism causes synchronization among nodes in their attempts to transmit. After they have been synchronized in the same slot, transmissions are delayed again, if nodes are in CS range, by the CSMA/CA method through carrier senses and back-offs. Conversely, when the Blind mechanism runs directly over CSMA/CA, transmissions occur soon after the MAC layer receives the message, instead of at the beginning of a synchronized TDMA slot.

When 2BSheld is applied over TDMA, it does not present significant performance gains compared to Blind-CSMA/CA scheduling, thereby questioning its usefulness. However, it is important to emphasize that flood control mechanisms do

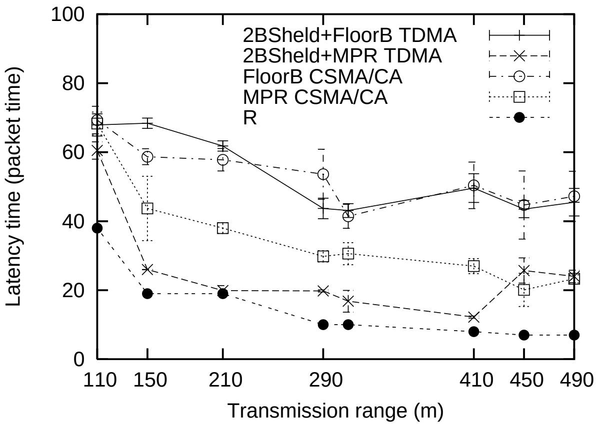
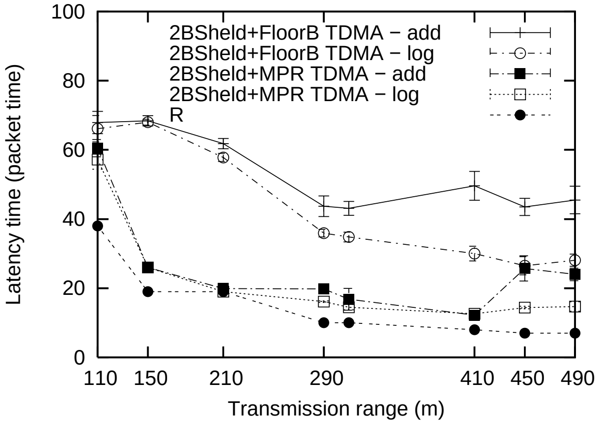
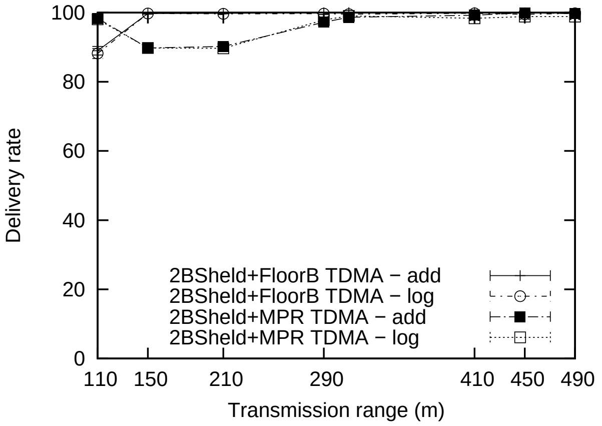


Fig. 4. Scheduler flooding latency joint to FloorB and MPR

not properly work over TDMA without a scheduler since the absence of carrier sense and back-off causes an excessive number of collisions. In some cases, messages do not go beyond the first hop.



(a) Flooding latency



(b) Delivery rate

Fig. 5. Latency and delivery rate using log function

2BSheld can be straightforwardly associated to FloorB and MPR since these two mechanisms use 2-hops neighborhood for their decisions. Results of this relationship are presented in Figure 4. One can observe no latency decrease when 2BSheld is associated with FloorB, but there is a large improvement with MPR. This occurs because for this type of scenario FloorB does not provide a significant reduction of forwarded messages, generating numerous messages to be scheduled. However, as MPR reduces the number of nodes to be scheduled, latency is decreased to values very close to R.

Nevertheless, for dense networks, i.e. for transmission ranges larger than $410m$, one can observe that 2BSheld cannot obtain, even when associated with MPR, the same approximation of theoretical bound R. This is due to the process of decision of slot allocation, which is performed adding the values $lowPri$ and $lowId$. Through this process, a node v can be scheduled up to N_1^v slots ahead, causing a large latency in dense networks, especially to reach edge nodes.

For this reason, we have evaluated the impact of the $s(x,y)$ function on the observed latencies. Figure 5(a) shows the flooding latency for two different empirical functions, the original additive function $s(x,y) = x+y+1$ and a logarithmic function described by $s(x,y) = round(2*log(x+y+1))+1$. The results show that log function provides lower latencies for dense networks than the additive one, possibly coming at the cost of an increase of scheduling conflicts but without affecting the delivery rate, as can be seen in Figure 5(b). The intuition behind the conception of this new function was to return 1 for $x = y = 0$ and to be monotonically increasing but providing lower values than the ones given by the additive function.

From the above observation, we propose a new criterion for slot allocation, i.e. a new function $s(x,y)$, based on the fact that as messages get "older" the faster they should be scheduled. This idea can be used to determine a time threshold for scheduling a message, which can be set by the flooding application according to R or, if it is unknown, in a certain number of slots. For example, if the flooding process is used to broadcast an updated node state information in a periodical basis, then it does not make sense to receive a message flooded in a previous period.

Therefore, we can formalize a new function $s(x,y)$ as follows. Let s be the number of slots from now until the schedule of an incoming message calculated by a receiver node r. One have that the greatest possible value of s is $s_{max} = |N_1^r|$. Let T be the time threshold for message expiration in number of slots. If $T \gg R$, t is the current slot, and h is the number of hops traversed by a message to reach the node r, one have (1) and (2).

$$s \leq T - t \tag{1}$$

$$s_{max} = T - t \tag{2}$$

However, from the point of view of r, after t slots the message has traversed h-hops. Considering this as the average displacement of the messages in the network, after T slots they should traverse the complete network diameter, i.e. R hops, whose proportionality can be established by the equation 3. Thus, rewriting s_{max}, one obtain 4.

$$T = \frac{Rt}{h} \tag{3}$$

$$s_{max} = \frac{Rt}{h} - t \tag{4}$$

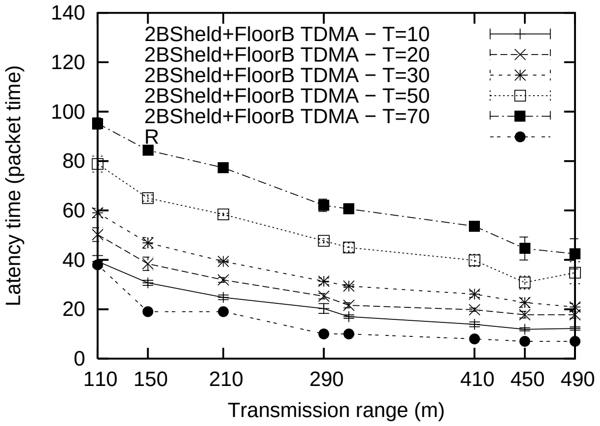
By establishing a ratio between s_{max} in (4) and its initial value equal to $|N_1^r|$, we can use the same proportionality between $s(lowPri, lowId) = lowPri + LowId + 1$ and the corrected value of s (s'). Then, one have (5) and (6).

$$s'(lowPri, lowId) = \frac{s(lowPri, lowId) \times (\frac{Rt}{h} - t)}{|N_1^r|} \tag{5}$$

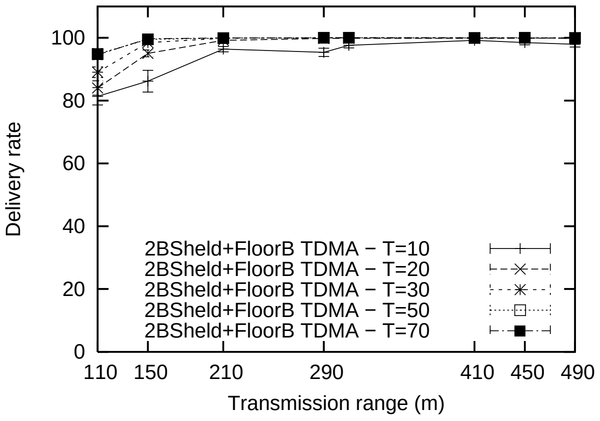
$$s'(lowPri, lowId) = \frac{(lowPri + lowId + 1)t(R - h)}{h|N_1^r|} \tag{6}$$

It is important to observe that, except for R, all other variables in these equations are obtained locally or from flooded messages. If R is not known, the equation (5) can be easily rewritten in terms of T (equation 7), which in many cases may be defined by the application that is performing flooding.

$$s'(lowPri, lowId) = \frac{(lowPri + lowId + 1)(T - t)}{|N_1^r|} \tag{7}$$



(a) Flooding latency



(b) Delivery rate

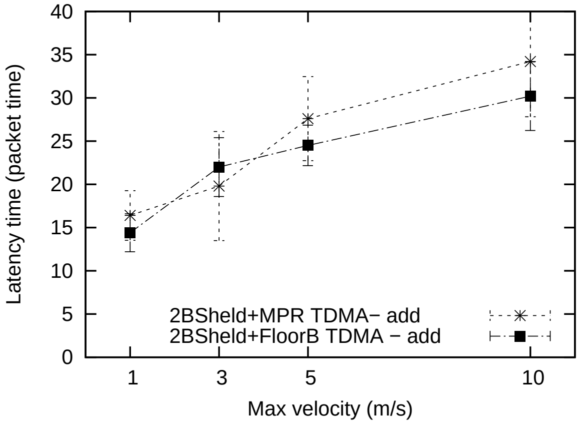
Fig. 6. Latency and delivery rate using T limit

Figure 6(a) shows latency as a function of the T limiter. One can see that latency is not limited to the configured ceiling. This is because in a slot s, the scheduler will distribute nodes in such a way that the last scheduled node will be allocated the slot $T - s$. However, for $s \geq T$, all scheduling will be done to the next slot. In the case that there is no conflict or collision, neighbors will receive the message and new schedules will be made in succession, but always to the next slot.

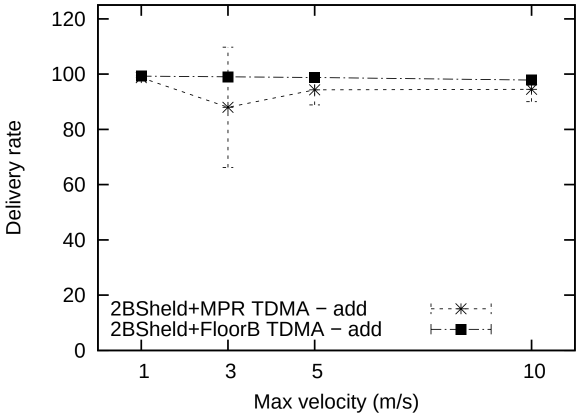
In Figure 6(b), one can observe the impact of the use of time limiter on the delivery rate. It allows decreasing the latency without a significant reduction in the delivery rate when $T \gg R$.

These scenarios also clearly show the influence of T limiter in total scheduling time, indicating that overestimated choices of T can lead to very long schedules.

As can be seen, the 2BSheld approach is simple and does not depend on the knowledge of network topology. In addition, the results demonstrate its efficiency in the used static scenarios. Following, 2BSheld is evaluated in mobile scenarios, which were not addressed by previous works.



(a) Latency

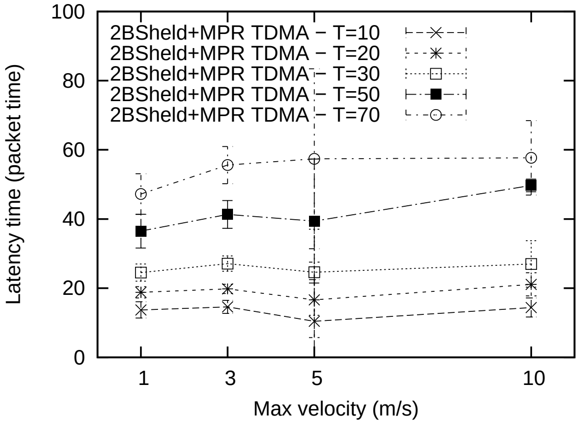


(b) Delivery Rate

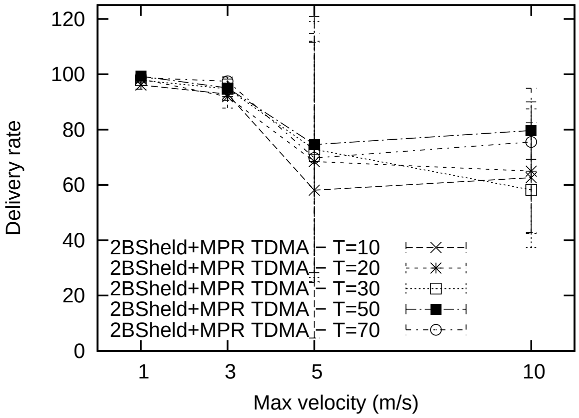
Fig. 7. Latency and delivery rate with mobility

In these scenarios, 200 mobile nodes are placed in a square area of $800m \times 800m$. These nodes use a transmission range of 120 meters and follow a Random Waypoint mobility model with maximum speeds that vary from 0 to $10m/s$. Figures 7(a) and 7(b) show the latency and the delivery rate as function of the nodes maximum velocity, respectively. These results were obtained with the scheduler associated with FloorB and MPR mechanisms, and the s additive function.

One can observe that latency is somewhat influenced by network mobility, being slightly smaller with FloorB when compared to MPR. In mobile scenarios, MPR is less efficient than FloorB in terms of delivery rate, which is a result already presented in [10] and independent of the scheduler. We also observe that some MPR results have presented large error bars. This occurs because in a few simulation runs the delivery rate is closer to 0% since messages did not surpass the first hops. In these scenarios, the sender node loses the neighborhood with MPRs nodes due to mobility and forwards a message before Hello messages could update the MPR mechanism.



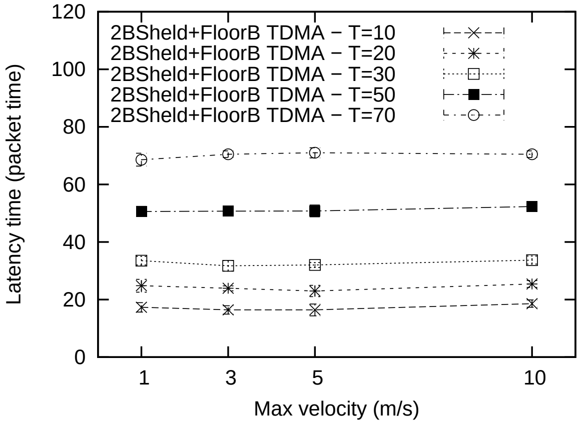
(a) Latency



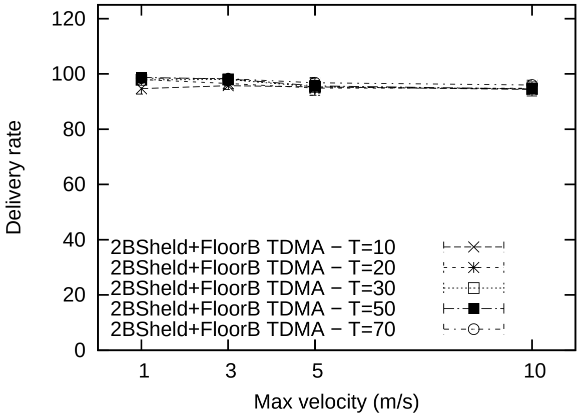
(b) Delivery Rate

Fig. 8. Latency and delivery rate - MPR with mobility and T limit

Figure 8 shows the performance of 2BSheld associated with MPR using T limit. In Figure 8(b), the delivery rate is almost independent of T, and decreases with mobility. Large error bars also appear at some velocities. Figure 8(a) shows that latency is directly proportional to T.



(a) Latency



(b) Delivery Rate

Fig. 9. Latency and delivery rate - FloorB with mobility and T limit

Conversely, Figure 9 shows 2BSheld performance when it is associated with FloorB and also uses a T limit. Results are similar to the ones with MPR. However, with FloorB, it is more stable against mobility effects than MPR, presenting very small error bars.

VI. CONCLUSIONS AND FUTURE WORKS

This paper presents an approach to deal with the Minimum Latency Broadcast Scheduling (MLBS) problem. Although this is a NP-complete problem tackled by several proposals in the literature, this work shows that approximation algorithms present performance bounds closer to the expected upper bound than to the lower. Also, the paper evaluates the latency produced by well-known flood control mechanisms, indicating that when conducting a proper choice of forwarding nodes, latency can be substantially reduced. Furthermore, it is shown that the expected latency for an optimal mechanism will have a value between R and N, and that depending on the topology, R takes values between 1 and $N - 1$.

From these observations, we propose a simple mechanism (2BSheld) based on a "greedy" algorithm, which provides a scheduling using solely local decisions. Although it does neither provide an upper bound nor avoid the occurrence of conflicts, it presents a good performance from sparse to dense networks, and also with variable nodes mobility. Moreover, the 2BSheld mechanism provides a way to control the total duration of network flooding, at the cost of small decreases in delivery rate.

We consider as the main contributions of this work the evaluation of the expected performance of some well-known flooding schedulers and the establishment of assumptions to conceive mechanisms completely distributed and based on local decisions. According to the results, one can identify that approximation solutions, even using global information or complex mechanisms, do not provide significant performance gains. In other words, despite being an NP-complete problem, simple heuristics allow to obtain results very close to the optimal solution in most scenarios.

The future work can be concentrated in two main directions. The first is to seek the integration between scheduler and flood control mechanisms, so that a mechanism decision will impact on other (cross-layer design). The second is to investigate the use of both the 2-hop neighborhood knowledge and the Equation 5 in the scheduling design.

References

[1] T. Clausen and P. Jacquet, “RFC 3626 - Optimized Link State Routing protocol OLSR,” *IETF Request for Comments*, Oct. 2003. [Online]. Available: ftp://ftp.rfc-editor.org/in-notes/rfc3626.txt

[2] C. Perkins, E. Belding-Royer, and S. Das, “RFC 3561 - Ad hoc On-demand Distance Vector AODV routing,” *IETF Request for Comments*, Jul. 2003. [Online]. Available: ftp://ftp.rfc-editor.org/in-notes/rfc3561.txt

[3] V. Lenders, M. May, and B. Plattner, “Service Discovery in Mobile Ad Hoc Networks: A Field Theoretic Approach,” in *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, Jun. 2005, pp. 120–130.

[4] C. H. P. Augusto and J. F. de Rezende, “An adaptive approach to service discovery in ad hoc networks,” in *8th IFIP IEEE International Conference on Mobile and Wireless Communications Network, 2006.*, Aug. 2006, pp. 61–75.

[5] G. Karumanchi, S. Muralidharan, and R. Prakash, “Information dissemination in partitionable mobile ad hoc networks,” in *Proceedings of IEEE Symposium on Reliable Distributed Systems (SRDS)*, Oct. 1999, pp. 4–13.

[6] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, “The broadcast storm problem in a mobile ad hoc network,” in *MobiCom ’99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*. New York, NY, USA: ACM, 1999, pp. 151–162.

[7] Z. Haas, J. Y. Halpern, and L. Li, “Gossip-based ad hoc routing,” in *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications*, vol. 3, Jun. 2002, pp. 1707 – 1716.

[8] A. Laouiti, A. Qayyum, and L. Viennot, “Multipoint relaying: An efficient technique for flooding in mobile wireless networks,” in *35th Annual Hawaii International Conference on System Sciences (HICSS’2002)*, Jan. 2002.

[9] V. Drabkin, R. Friedman, G. Kliot, and M. Segal, “Rapid: Reliable probabilistic dissemination in wireless ad-hoc networks,” in *SRDS ’07: Proceedings of the 26th IEEE International Symposium on Reliable Distributed Systems*. Washington, DC, USA: IEEE Computer Society, Oct. 2007, pp. 13–22.

[10] C. H. P. Augusto and J. F. de Rezende, “Floorb: Mecanismo de controle de inundao para redes ad hoc mvel,” *Revista Brasileira de Redes de Computadores e Sistemas Distribudos (RESD)*, vol. 2, no. 1, pp. 35–49, Jun. 2009.

[11] T. Fujie, “An exact algorithm for the maximum leaf spanning tree problem,” *Computers and Operations Research*, vol. 30, no. 13, pp. 1931 – 1944, 2003. [Online]. Available: http://www.sciencedirect.com/science/article/B6VC5-47RRRSW-3/2/62e2041b62c7c0a80171b40b51e79c5a

[12] R. Gandhi, A. Mishra, and S. Parthasarathy, “Minimizing broadcast latency and redundancy in ad hoc networks,” *IEEE/ACM Trans. Netw.*, vol. 16, no. 4, pp. 840–851, 2008.

[13] S. C.-H. Huang, P.-J. Wan, X. Jia, H. Du, and W. Shang, “Minimum-latency broadcast scheduling in wireless ad hoc networks,” in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, May 2007, pp. 733–739.

[14] L. Badia, A. Erta, L. Lenzini, and M. Zorzi, “A General Interference-Aware Framework for Joint Routing and Link Scheduling in Wireless Mesh Networks,” *IEEE Network*, vol. 22, no. 1, pp. 32–38, Jan. 2008.

[15] D. J. Vergados, M.-Y. Manolaraki, and D. D. Vergados, “Evaluation of broadcast scheduling algorithms for ad-hoc tdma networks,” in *1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology, 2009. Wireless VITAE 2009.*, May 2009, pp. 394 – 398.

[16] X.-Z. Zhang, “Efficient broadcast scheduling based on fuzzy clustering and hopfield network for ad hoc networks,” in *International Conference on Machine Learning and Cybernetics, ICMLC 2007*, vol. 6, Aug. 2007, pp. 3255–3260.

[17] X. Wu, B. Sharif, O. Hinton, and C. Tsimenidis, “Solving optimum tdma broadcast scheduling in mobile ad hoc networks: a competent permutation genetic algorithm approach,” *Communications, IEE Proceedings-*, vol. 152, no. 6, pp. 780–788, Dec. 2005.