Mobility-aware Adaptive Counter-based Forwarding Elimination to Reduce Data Overhead in Multicast Ad Hoc Routing

Carmen M. Yago Sanchez, Pedro M. Ruiz and Antonio F. Gomez Skarmeta
Dept. of Information and Communication Engineering, University of Murcia,
Campus de Espinardo,
30100 Murcia, Spain
Email: {carmen,pedrom,skarmeta}@dif.um.es

***Abstract*— Most of the previous efforts regarding multicast routing in MANETs, have been devoted to the provision of low-control overhead protocols, being able to maximize the packet delivery ratio. In multicast routing, the non-optimality of the forwarding structure can also lead to transmission of additional data packets (compared to the minimum required). We call those additional data packets data-overhead. In this paper, we present a counter-based forwarding elimination scheme, being able to reduce that overhead depending upon the mobility of the nodes. Our results show that this approach is able to enhance the bandwidth consumption of mesh-based multicast ad hoc routing protocols while maintaining nearly the same packet delivery ratio.**

I. Introduction and Motivation

Nowadays mobile and wireless technologies are responding to the necessity of communicating everyday and everywhere without restrictions. In this atmosphere, ad-hoc networks, the infrastuctureless wireless networks, are gaining momentum. The ad-hoc networks (MANETs) consist of a group of mobile hosts communicating among them without any fixed infrastructure. Nodes auto-configure themselves to set up a network where every node can be both a host and a router so that any message can be delivered to its destination either directly or via multihop paths.

These networks are creating a big interest due to the variety of applications they have in different environments, for example rescue operations, battlefields, or communication between home automation devices.

There is also a manifest interest in allowing these networks to make use of multicast communication. Several routing protocols have been developed to route multicast traffic. These protocols are basically grouped into three categories [1]: stateless multicast like DDM, tree based protocols like MAODV or AMRIS and mesh based protocols like ODMRP or CAMP. There are also some hybrid approaches like AMRoute.

Stateless multicast protocols are oriented to small groups. Tree based protocols do not have this restriction but they usually have problems with high mobility networks. In that case, they are outperformed by mesh based protocols which introduce more redundancy and alternate paths.

Mesh based approaches seem to be a good way to route multicast traffic in ad hoc networks, however they produce a considerable overhead. This overhead has two causes: firstly, the instability of the network makes it necessary to flood control messages periodically. Secondly, there is data overhead. Data overhead is a consequence of the non-optimality of multicast trees and meshes. Multicast meshes provide robustness but also cause forwarding nodes to redundantly transmit the same message in the same area. Then, as defined in [2] this data overhead consists of the data messages unnecessarily transmitted due to the redundancy of the mesh. In addition, provided that data traffic rates are higher than control packet rates, data overhead becomes the main source of sub-optimality of routing protocols, producing excessive bandwidth consumption, increased link-layer contention, and most of the issues associated to blind flooding [3].

Ruiz [2] showed that the problem of computing the minimal data-overhead multicast tree is an NP-complete problem. So, we consider that a probabilistic approximation algorithm based on the idea of limiting the number of redundant data messages without pruning multicast meshes will improve the efficiency whilst keeping the robustness. One of the methods which fulfills this idea is the counter-based algorithm defined in [3]. Following this approach we propose the mobility-aware counter-based mechanism, a new enhanced version which adapts itself to network conditions.

This paper is organized as follows, in the next section we describe a brief study of proposals which exist in the literature to reduce overhead in both flooding and multicast traffic. Then in Section 3 we propose the adaptation to multicast of the counter-based mechanism and a way to improve it. In Section 4 we present and evaluate the results of the simulation. Finally, the conclusions are provided in the last section.

II. Related Work

Several algorithms and protocols have been developed to limit the number of messages sent into the network without losing reliability. The proposals to give more efficiency to traffic delivery in ad-hoc networks can be basically grouped into two categories [4], topology based approaches and heuristic based approaches. The former refers to the use of topological information to reduce the number of nodes which can retransmit a message. The latter, rather than reducing the number of nodes which can retransmit, allows every node which is in charge of forwarding the message to decide, using a heuristic, whether to retransmit it or not.

There are several approaches following the topology-based concept. Among them, there are neighbour-topology-based schemes such as self-pruning [5] or multipoint relay MPR [6] which use the information of one-hop or two-hop-away neighbours to decide which node is a forwarder. There are also hierarchical approaches such as Domain-Based [7] or Overlay-Driven hierarchical multicast in which nodes are partitioned in hierarchical sub-groups and the data is delivered from the upper level subgroup to the lower.

In the heuristic category Ni *et al.* defined in [3] three schemes: counter-based, where a node does not rebroadcast if it has heard the duplicated messages more than a number of times; distance based, where the node decides whether to resend or not depending on its distance to other nodes; and probabilistic, where the node makes a decision based on a probabilistic function. For example Source Grouped Flooding for multicast routing [8] uses a probabilistic scheme to reduce redundant data transmissions. This category is useful to deal with the inherent instability of MANETs, since there is no reduction of the number of alternative paths.

Within this category we consider the counter-based algorithm particularly remarkable, due to its performance, its ease of calculation and low resources consumption. It is based on the ”expected additional coverage” concept. Every time a message is retransmitted by a node it covers an area in which the message can be heard. The ”expected additional coverage” is the new area that would be covered for the first time if the node retransmitted the message. This area gets smaller every time a node listens to the message it has to retransmit, because part of this area is already covered by the message heard. In fact, the expected additional coverage, after having listened to the same message for three times is below 10% [3].

This concept is illustrated in the example shown in Fig. 1: nodes A and B forward the same message which is heard by nodes C and D. C receives the message twice (from both A and B) and D only once (from B). In both cases the ”expected additional coverage” is the non-shadowed part of its communication range, because every node in their shadowed area has already heard the message from nodes A, B, or both. Thus, if C or D rebroadcast the message, only the nodes in the non-shadowed area will receive the message for first time. The ”expected additional coverage” of C is smaller than the ”expected additional coverage” of D because the communication range of C has been almost totally covered by the communication ranges of A and B, while the communication range of D is only partially covered by the communication range of B.

The counter-based algorithm proposes that a message must not be retransmitted if the host has heard it k times before, when the ”expected additional coverage” becomes too small.

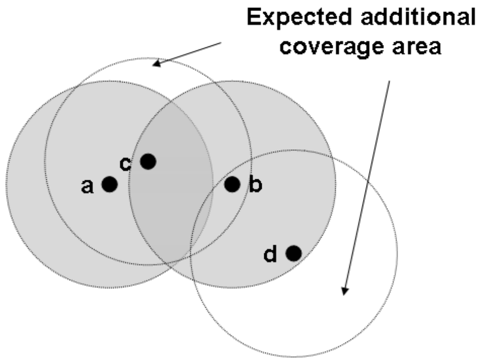

Fig. 1. Example of the expected aditional coverage concept

Following the example in Fig. 1 and considering for simplicity k=2, D will rebroadcast the message because it only has received the message once and C will not resend the message because it has received the same message twice, and so it considers its ”expected additional coverage” area has become small enough.

There are also emerging adaptive proposals which adapt these algorithms to some characteristics of the network such as the density of neighbours. For example, following this metric, in [9] the Border Node Retransmission Based Probabilistic Broadcast Protocols are defined, and in [10] an adaptive counter-based scheme is proposed.

In spite of the fact that adaptive ad hoc routing protocols may overtake the performance of non-adaptive ones, a representative metric is necessary. Boleng, Navidi, and Camp defined in [11] the requirements for a mobility metric to serve as the basis for an adaptive protocol. This metric has to be:

- Computable in a distributed environment.
- Good indicator of protocol performance.
- Feasible to compute.
- Independent of any specific protocol.
- Computable in a real network.

Based on this, we propose here a method to reduce the data overhead in mesh based protocols: the mobility-aware counter-based algorithm. It is a new adaptation of the counter-based algorithm which adapts itself to network conditions using a new mobility metric which fits the requirements detailed above. This metric, the modal link duration interval, is based on the stability of the links and allows the node to be aware of its local network conditions.

III. Mobility-aware Counter Scheme

As we have seen before, counter-based mechanisms [3], [10] have been revealed as a good approach to reduce the redundancy in flooding. We use a similar approach for dealing with data overhead reduction. Every node belonging to the mesh has a counter c. When a node has to forward a message, it sets a timer and, for a period of time, increments the counter every time it hears the same message it has to send. When the timer expires, if the counter c has reached a threshold value C, the data message is not retransmitted, but silently discarded.

As said in Section 2, the justification of this behaviour resides in the ”expected additional coverage” concept. That is the area where a message can be heard for the first time as a result of the node having forwarded it. Every time a node listens to a message this area gets smaller because part of it is already covered by the received message. Then, when a message has been heard a determined number of times, this new area is so small than it usually makes no sense to retransmit the message.

Using this counter-based approach, the threshold value C, is the number of messages listened to in which the expected additional coverage is so small that it is worthless to introduce more redundancy. This value has to be carefully chosen and it has to be a trade off between the performance of the protocol and the reduction of the overhead. A greater threshold provides better performance, but also higher overhead.

Moreover, when the links are instable the number of messages heard has less significance because the instability of the links may have prevented the other nodes in the area from receiving the message.

This might be solved by using a greater fixed threshold C, but then the message saving would decrease with the consequent increment of the overhead and its problems.

Due to these disadvantages, we consider the traditional counter-based scheme can be improved by making every forwarder node vary its threshold C dependant on its local network conditions. To do this, we need an adaptive and distributed metric which tells the node about the stability of its links. If the links are stable, the messages the node is listening to, have probably been already received by the other nodes in the area. Below, we define our proposed metric, and afterwards we will describe our proposed approach.

A. Modal link duration interval (MLD) mobility metric

Boleng, Navidi, and Camp [11] have defined the link duration metric to be computed by a node in the following way: for each neighbour, the average life of the links they have is computed. Then the link duration is the mean of all of these averages. They showed the link duration as a mobility metric which is a good indicator for protocol performance. This is because it is computable in a distributed environment, feasible to compute, independent of any specific protocol and computable in real network implementations.

This metric fits many of our requirements: it is distributed and reflects network conditions. But if a few nodes behave very differently from the rest, the average may not reflect the behaviour of the majority of the network. We desire a metric which reflects the behaviour of the majority of the forwarder neighbours, hiding the distortion produced by ”rebel” nodes. Thus, instead of the mean, we have adopted a modal interval of the average link duration of every forwarder neighbour. The way we compute the metric is the following: firstly, we split the set of real numbers into intervals (for example, in Fig. 3, the x axis has been split into three intervals). Secondly, for a period of time T, every node computes the average link duration LD_f with each forwarder neighbour it has. In order to compute if the links are up or down, the period of time T is divided into k timeslots whose size t has to be long enough to allow the node to receive at least one message from every neighbour (this happens at least every periodic flooding timeout). Then, the average link duration for each forwarder node f in a period of time T, LD_f is defined as:

- $h(i)_f$ function determines if the link with f is alive in the timeslot i.

$$h(i)_f = \begin{cases} 1 \text{ if the node has heard a message from f} \\ 0 \text{ otherwise} \end{cases} \tag{1}$$

- Ch_f calculates how many times the link goes up in the period T.

$$Ch_f = \begin{cases} \sum_{i=1}^{i=k-1} \overline{h(i)} * h(i+1) \text{ if } h(1) = 0 \\ \\ 1 + \sum_{i=1}^{i=k-1} \overline{h(i)} * h(i+1) \text{ if } h(1) \neq 0 \end{cases} \tag{2}$$

- Then LD_f is calculated by dividing the time in which the link with node f is alive over the times the link goes up in the period T:

$$LD_f = \begin{cases} \sum_k h(i)_f \text{ if } Ch_f = 0 \\ \\ \frac{\sum_k h(i)_f}{Ch_f} \text{ if } Ch_f \neq 0 \end{cases} \tag{3}$$

Thirdly, every computed LD_f belongs to an interval from the set previously defined in the first step, then, modal link duration interval MLD is the interval to which the greatest number of LD_f, belongs.

In order to adapt the metric asymptotically to the changing network conditions, the T period has a window structure. The metric is computed every t time units. That is, if we compute the metric for a period T which lasts from an instant m to an instant $m + k * t$, in the following computacion of the metric, the period T will last from the instant $m + t$ to the instant $m + (k + 1) * t$.

This metric gives the node an idea of what is happening with the majority of the nodes in the neighbourhood, what is happening in the area it is in. For example, if MLD is higher, the network around is basically stable, no matter if there are a few nodes with low stable links or they are moving together in the same direction.

B. The Mobility-aware Counter-based Algorithm

The mobility-aware counter-based algorithm used to reduce data overhead in mesh based protocols is built around the basic counter algorithm. There are two processes working concurrently: the first computes the value of the threshold C as a function of MLD, $C = CC(MLD)$, whereas the second applies the counter-based approach. Fig. 2 shows how the mobility-aware counter-based algorithm works.

We have chosen a function $C = CC(MLD)$ with the shape shown in Fig. 3 based on the following heuristics:

- If the value of MLD is low, the node itself has not been able to establish stable links with other nodes. Probably,

```
mainLoop()
{
    msg = receiveMessage();
    if(msg.isDuplicate() == false)
    {
        msg.setCounter(1);
        msg.setTimer(random(0,... tmax));
        msg.startTimer();
    }
    else
    {
        if(msg.timerExpired() == false)
            msg.incrementCounter();
    }
    ...
}
/* Event handler called when timer expires
for message 'msg' */
timerExpired(msg)
{
    C=CC(MLD);
    if(msg.getCounter() == C)
        msg.discard();
    else
        msg.retransmit();
}
```

Fig. 2. Pseudocode for the mobility-aware counter-based algorithm

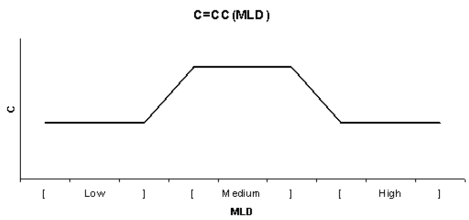

Fig. 3. Shape of the CC(MLD) function

when it is going to resend the message, the links with the intended receivers will be broken. For this reason, it will only retransmit the message if it has heard it very few times, to deal with the possibility that there are hardly any nodes in the area able to forward the message.

- If the value of MLD is medium, the node is in a network where a great number of the nodes are able to set links of a moderate duration. A medium value of C helps to cover the expected area.
- If the value of MLD is high, the network is basically stable: there are few changes, so a low threshold should be enough.

IV. Simulation Results

The data overhead reduction mechanisms presented above are going to be applied to a multicast routing protocol to test if they really assess the performance of the proposed approach. We have chosen ODMRP because it is a well known mesh based protocol and offers a good performance when compared with other multicast routing protocols [12] not only tree based protocols like AMRIS or hybrid approaches like AMRoute but also mesh based ones like CAMP.

We have made some modifications to the ODMRP protocol to introduce these data overhead reduction mechanisms: all multicast forwarder nodes execute one of the counter-based algorithms before forwarding a data message. Moreover, when mobility-aware mechanisms have been used, the multicast forwarders have a process which computes the modal link duration interval to apply the heuristic to establish the threshold C. No other modifications have been carried out.

To simulate our mobility-aware counter scheme we have considered the following time ranges:

- MLD is the interval $[0, 21)$: it has a low value. Then C is low, and we considered low $C = 2$.
- MLD is the interval $[21, 75)$: it has a medium value. Then C is medium and we considered medium $C = 3$.
- MLD is the interval $[75, \infty)$: it has a high value. Then C is low, and we considered low $C = 2$.

The first time range which defines low stable links is defined according to [11] which shows that with links with a lifetime lower than 20 seconds, the instability of the network leads to a poor performance of the protocols. The range which defines long lived links has been set because we estimate that links which last more than 75 seconds are highly stable links when dealing with the instability of ad-hoc networks.

The period of time T in which the MLD is calculated is 90 seconds which we consider a good value to capture both long and short lived links. The MLD is computed every 3 seconds because that is the flooding timeout.

We have chosen $C = 3$ as the medium value because according to [3], the additional expected coverage is about 10%, which means that when a node has heard the message 3 or more times, the nodes which are in the 90% of its communication range have already received the message and thus, a new retransmission would be redundant for all of them.

We have simulated the different approaches in the NS-2 [13] network simulator. We have used the version 2.1b8 with the multicast extensions developed by the Rice University Monarch Project [14] which include the ODMRP implementation used here.

The simulated scenario consists of 100 mobile hosts randomly distributed over an area of 1600x1200m. The radio channel capacity for each mobile node is 2 Mb/s, using the IEEE 802.11b DCF MAC layer and a communication range of 250 m. Each one of the approaches has been evaluated over the same pre-generated set of 330 scenarios with varying mobility speed and traffic loads. Mobile nodes move using a Gauss-Markov model [15]. In this model, initially each node is randomly assigned a speed and a direction. This speed and direction is changed at discrete time intervals. We have chosen a maximum speed of 0, 5, 10, 15, and 20 m/s. The velocity is updated every 10 seconds, the angle standard deviation is $\pi/8$ and the speed standard deviation is 0.1. Ten different traffic loads where tested consisting of 1, 2, and 4 CBR sources for the same multicast group, and a varying number of receivers, 5, 15, and 30.

Ten different traffic loads where tested consisting of 1, 2, and 4 CBR sources for the same multicast group, and a varying number of receivers, 5, 15, and 30.

A. Performance Metrics

To assess the effectiveness of the different mechanisms, we have used the following performance metrics:

- Packet delivery ratio. Defined as the number of data packet successfully delivered over the number of data packets generated by the sources.
- Normalized packet overhead. Defined as the total number of control and data packets sent and forwarded normalized by the total number of packets successfully delivered.
- Forwarding Efficiency. The average number of times that a multicast data packet was forwarded by the routing protocol. This metric represents the efficiency of the underlying forwarding structure.
- Average delivery delay. For each receiver, the average delay of all packets received is computed. Then the average delivery delay is the mean of all of these averages.

B. Analysis of the Results

We have simulated three approximations of the counter-based scheme: two with a fixed threshold value of C=2 and C=3 and one with our mobility-aware counter approach. Their results have been compared with the results offered by the ODMRP protocol. Fig. 4 to Fig. 15 show the results with 15 receivers and 1, 2, and 4 sources. The increment of the number of sources implies the increase of both the traffic and the density of the forwarder nodes.

Fig. 4, Fig. 5, and Fig. 6 show the packet delivery ratio (PDR) as a function of the maximum speed of the nodes. In all simulations, the fixed threshold $C = 2$ offers an insufficient PDR: almost always under 95%. This is because having fewer retransmissions affects the network connectivity. Regarding the other approaches, counter-based schemes have worse PDR than ODMRP in sparse networks. This is because the mesh has not enough redundancy; this situation gets worse as speed increases because there are more link breakages and there are no alternate paths. As the network becomes denser, counter-based mechanisms offer a higher PDR and with $C = 3$, the speed only causes a slight drop. However, our mobility-aware counter scheme gets a higher PDR when the speed is higher than 10 m/s. This is because our scheme adapts itself to higher speeds.

Fig. 7, Fig. 8, and Fig. 9 show the normalized overhead as a function of the maximum speed of the nodes. In all cases, the use of counter-based schemes reduces considerably the overhead. The counter-based scheme which provides higher savings is $C = 2$ but as said before, its performance is too low. $C = 3$ provides the smallest savings (compared with ODMRP the saving is between 17% and 42%). The higher saving is achieved as more redundancy is in the network, but there is almost no variation with the increase of the speed. This is because with $C = 3$ there is enough redundancy. Our mobility-aware counter scheme provides better savings than $C = 3$ but worse than $C = 2$ (when compared with ODMRP the saving is between 27% and 56%). It also provides better overhead reduction when the mesh is denser. However there is a variation with the speed of the nodes: the saving is better in low mobility environments where the links are stable. This is because our scheme detects the stability of the network, and hence it can provide better savings.

Fig. 10, Fig. 11, and Fig. 12 show the forwarding efficiency (FEF) as a function of the maximum speed of the nodes. It is observed that the number of times a packet has to be forwarded experiments an important decrease compared with ODMRP when using counter-based schemes. This is because they prevent the nodes from retransmitting unnecessary broadcasts. Results are similar to those for overhead, and our mobility-aware counter scheme provides lower FEF than $C = 3$ but greater than $C = 2$.

Fig. 13, Fig. 14, and Fig. 15 show the average delivery delay in milliseconds as a function of the maximum speed of the nodes. Here, all counter-based schemes present similar results higher than the ones presented by ODMRP. This is because ODMRP always follows a shortest path tree approach, but when using a counter-based approach the path that a message follows from its source to its destination does not have to be the shortest one, since there are nodes which do not retransmit.

Summing up, these graphs show that counter schemes can be applied successfully to multicast mesh based protocols. In general, all the adaptations save overhead and get a better forwarding efficiency. It can also be noticed than this scheme works better in environments where the density of forwarder nodes is higher. Regarding fixed threshold approaches, when using $C = 3$, the PDR is similar to ODMRP. There is also an important reduction of the overhead. Yet, the threshold $C = 2$ has resulted insufficient because in spite of providing better overhead reduction (it reaches even a reduction of 60% respecting ODMRP) it offers a poor PDR, getting under 90%. This PDR is especially poor when dealing with high speed nodes which cause it to drop to 86%.

Regarding our mobility-aware counter scheme, its performance can be situated into both thresholds: it usually offers an acceptable PDR (near the PDR of the $C = 3$ approach) with a better forwarding efficiency and overhead reduction. In general the overhead reduction is between 10% and 25% better than with a fixed threshold of $C = 3$. The overhead curve is nearer $C = 2$ when the mobility is low because links last more time and it is nearer $C = 3$ when the mobility is high because of link breakages. The best performance is when the mobility is low but not completely static, in which it offers really better PDR than $C = 2$ but the overhead saving is almost the same.

V. Conclusions

The alternate paths grant robustness to multicast mesh based protocols at the expense of adding redundant data transmissions. Providing more efficiency to data dissemination has to be done while keeping the robustness. To address this question we have made the mesh nodes use the counter-based

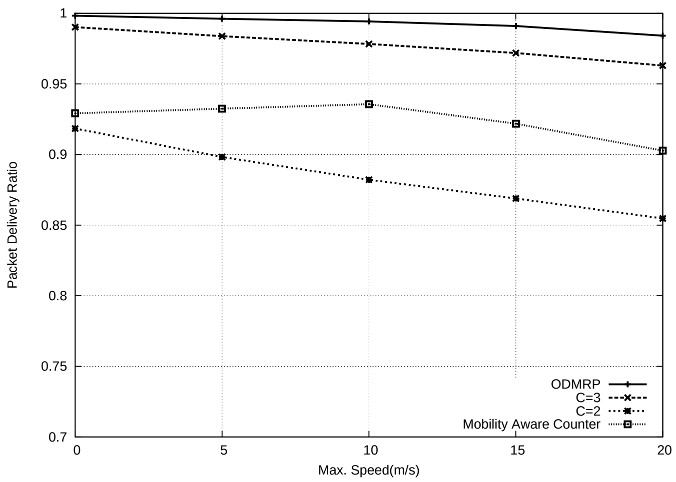

Fig. 4. PDR with 1 source and 15 receivers

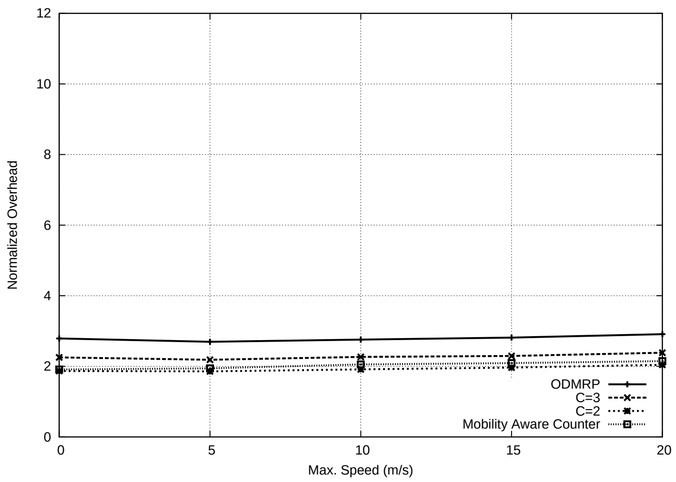

Fig. 7. Overhead with 1 source and 15 receivers

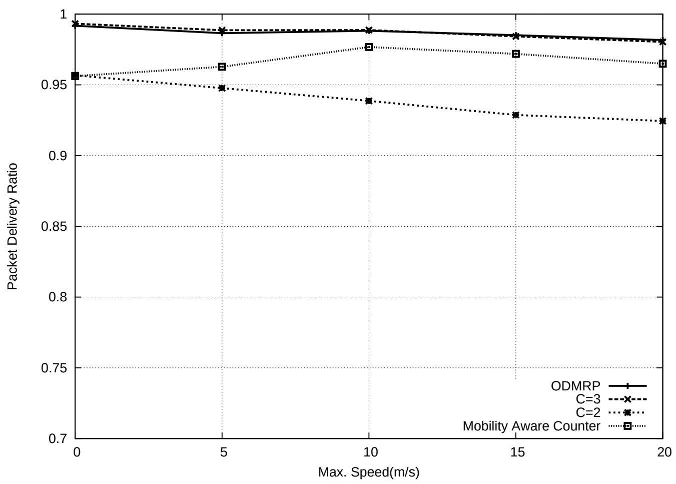

Fig. 5. PDR with 2 sources and 15 receivers

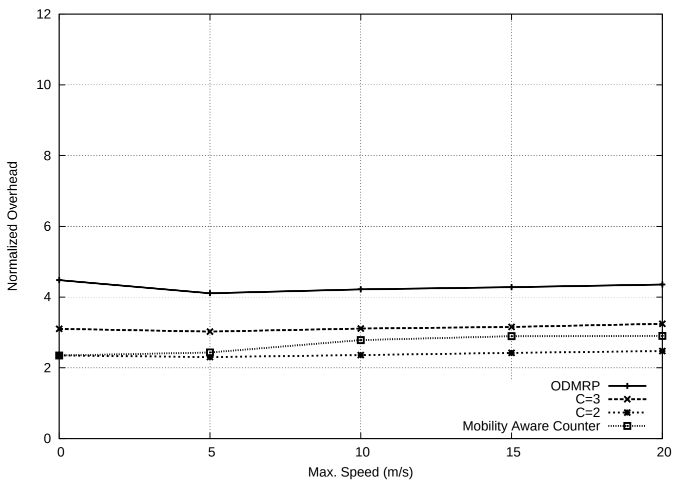

Fig. 8. Overhead with 2 sources and 15 receivers

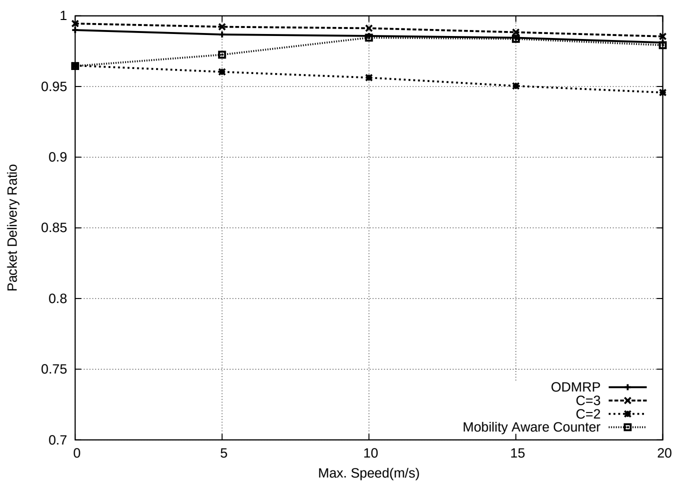

Fig. 6. PDR with 4 sources and 15 receivers

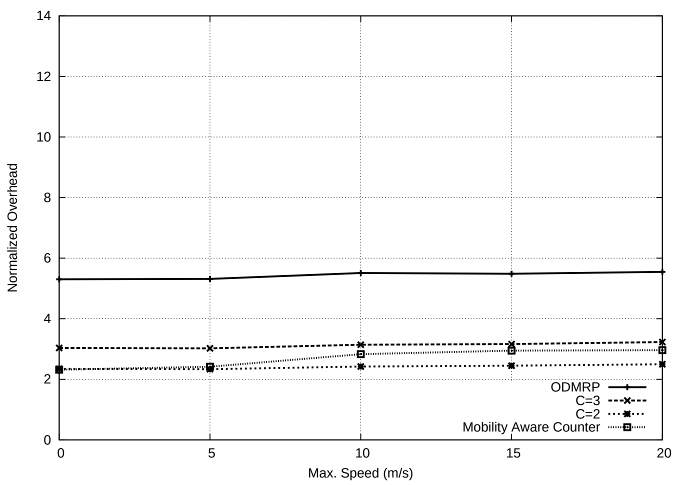

Fig. 9. Overhead with 4 sources and 15 receivers

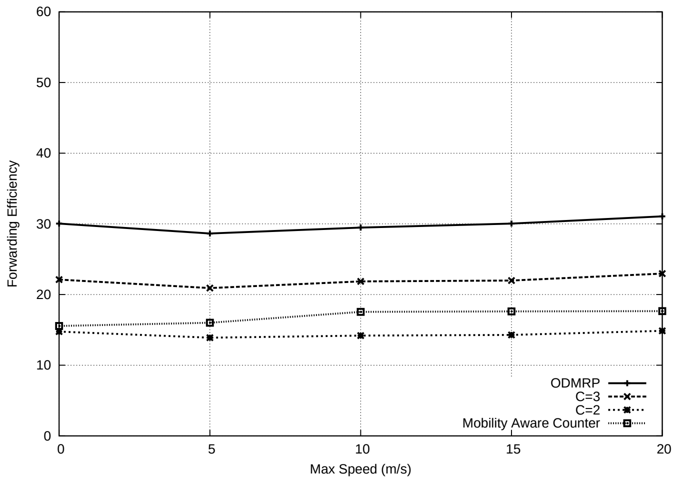

Fig. 10. FEF with 1 source and 15 receivers

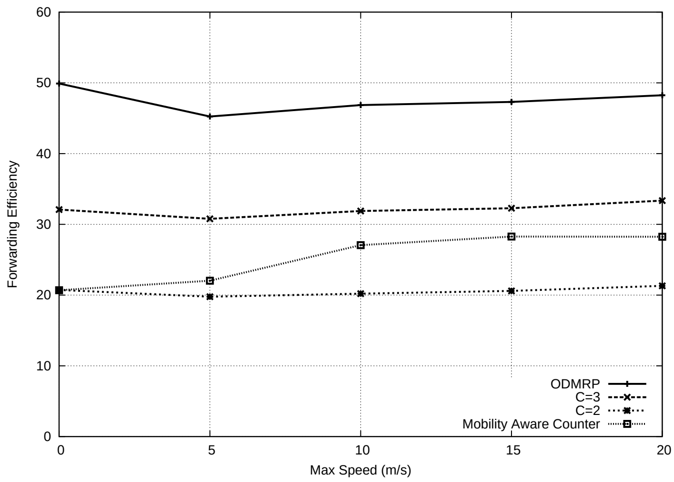

Fig. 11. FEF with 2 sources and 15 receivers

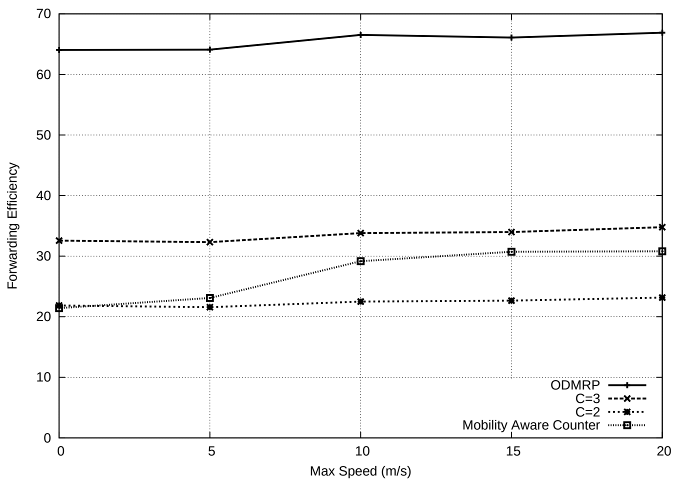

Fig. 12. FEF with 4 sources and 15 receivers

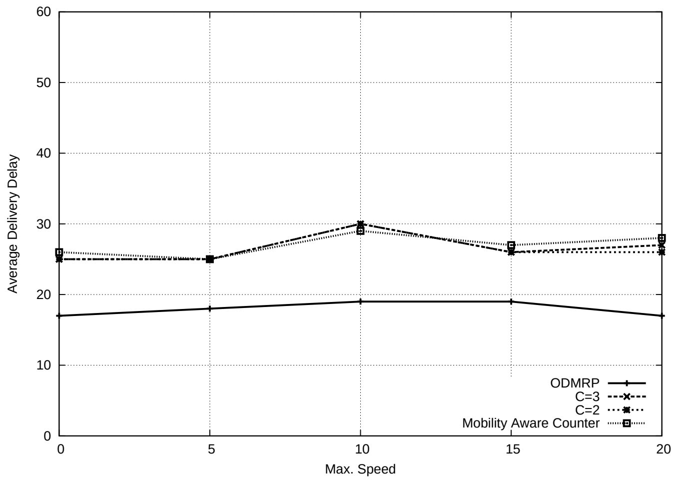

Fig. 13. Average Delivery Delay with 1 source and 15 receivers

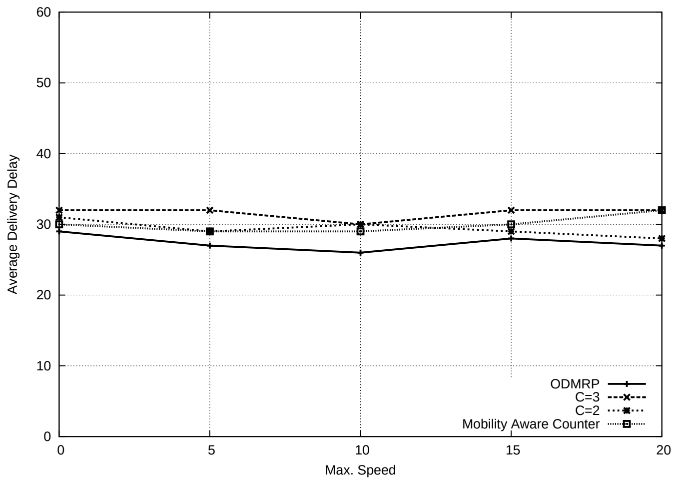

Fig. 14. Average Delivery Delay with 2 sources and 15 receivers

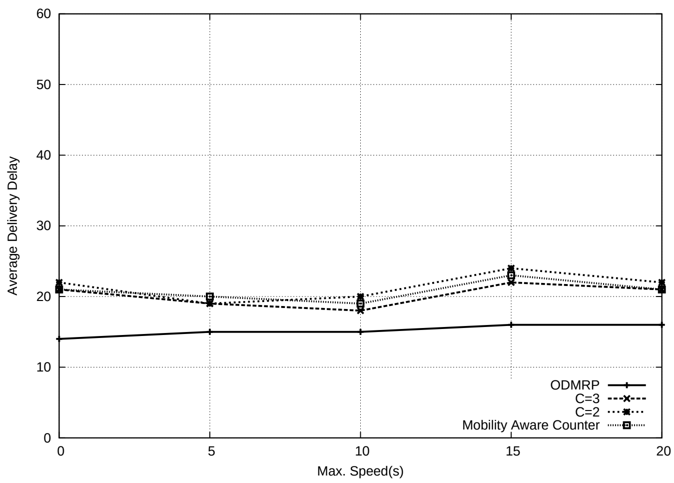

Fig. 15. Average Delivery Delay with 4 sources and 15 receivers

algorithm to decide whether to forward or not. This approach has proven to be a good solution to reduce data overhead while maintaining the performance. However, this approach is based on a fixed threshold value: if this value is low, the algorithm provides better savings at the expense of performing worse, and vice versa. For this reason we have proposed in this paper an adaptive variant of this approach. The proposed mobility-aware counter-based mechanism obtains good performance as high thresholds do, but has better efficiency in terms of bandwidth consumption.

Using the mobility-aware counter-based mechanism, every forwarder node changes its threshold value according to the stability of the network around it. To allow the node to realize the network conditions, the algorithm uses a new metric, the modal interval of the link duration, which gives the modal interval of the average duration of the links the node has. This gives the node an idea of how the majority of the nodes around it are behaving.

Using our mobility-aware counter-based mechanism, the mesh is not pruned, and in consequence,its robust structure is preserved. The good performance of the protocol is preserved and the overhead is reduced up to a 56%.

ACKNOWLEDGMENT

This paper has been partially funded by the Spanish MCYT by means of the SAM (TIC2002-04531-C04) project, and the ”Ramon y Cajal” workprogramme and also by the ”Consejeria de Trabajo, Consumo y Politica Social de la Region de Murcia” and the European Social Fund by means of the ”Integrated Operacional Programme for the Region de Murcia 2000-2006”.

REFERENCES

[1] C. M. Cordeiro, H. Gossain, and D. P. Agrawal, “Multicast over wireless mobile ad hoc networks: present and future directions,” *IEEE Network*, Jan./Feb. 2003.

[2] P. M. Ruiz and A. F. Gmez-Skarmeta, “Mobility-aware mesh construction algorithm for low data overhead in multicast ad hoc routing,” *Journal of Communications and Networks (JNC)*, vol. 6, no. 5, pp. 331–342, Dec. 2004.

[3] S. Ni, Y. Tseng, Y.S., Chen, and J.-P. Sheu, “The broadcast storm problem in a mobile ad hoc network,” in *Proc. of Int’l Conf. Mobile Computing and Networking (MOBICOM)*, August 1999, pp. 151–162.

[4] Y. Yi, M. Gerla, and T. J. Kwon, “Efficient flooding in ad hoc networks: a comparative performance study,” in *Proc. of the IEEE International Conference on Communications (ICC 2003)*, May 2003.

[5] H. Lim and C. Kim, “Multicast tree construction and flooding in wireless ad hoc networks,” in *Proc. of the 3rd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, 2000, pp. 61–68.

[6] 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’2001)*. IEEE Computer Society, 2001.

[7] C. Gui and P. Mohapatra, “Scalable multicasting in ad hoc networks,” in *IEEE INFOCOM*, 2004.

[8] K. Chandrashekar and J. Baras, “Multicast routing in mobile ad hoc networks using source grouped flooding,” *5th European Wireless Communications 2004*, Feb. 2004.

[9] J. Cartigny and D. Simplot, “Border node retransmission based probabilistic broadcast protocols in ad-hoc networks,” in *Proc. of the 36th Hawaii International Conference on System Sciences*, 2002.

[10] Y. Tesng, S. NI, and E. Shih, “Adaptive approaches to relieving broadcast storms in a wireless multihop mobile ad hoc network,” *IEEE Trans. Comput.*, vol. 52, no. 5, May 2003.

[11] J. Boleng, W. Navidi, and T.Camp, “Metrics to enable adaptive protocols for mobile ad hoc networks,” in *Proc. of the International Conference on Wireless Networks (ICWN ’02)*, June 2002, pp. 293–298.

[12] S. J. Lee, M. Gerla, and C. Chiang, “On demand multicast routing protocol,” in *Proc. of the IEEE WCNC’99*, Sept. 1999, pp. 1298–1302.

[13] The network simulator ns-2. [Online]. Available: http://www.isi.edu/nsnam/ns/

[14] The rice university monarch project. [Online]. Available: http://www.monarch.cs.rice.edu/

[15] T. Camp, J. Boleng, and V. Davies, “A survey of mobility models for ad-hoc network research,” *Wireless Communications and Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, vol. 2, no. 5, pp. 483–502, Sept. 2002.