A Novel Forwarding/Dropping Decision Engine for Wireless Multi-hop Ad-hoc Networks

A. M. AbdelRahman
Graduate Teaching Assistant
Dept. of Computer Engineering
Arab Academy for Science and Technology (AAST).
Alexandria, Egypt
Email: abdelrahman.aast@gmail.com

M. Abou El-Nasr
Associate Professor
Dept. of Computer Engineering
Arab Academy for Science and Technology (AAST).
Alexandria, Egypt
Email: mnasr@vt.edu

Ossama Ismail
Professor
Regional Informatics Center
Arab Academy for Science and Technology (AAST).
Alexandria, Egypt
Email: ossama@vt.edu

***Abstract*—Ad-hoc networks have been massively used in the last couple of years. The noticeable increase of wireless devices including PDAs, mobile phones, cellular devices, notebooks, etc has directed researchers to start considering various kinds of such ad-hoc networks including wireless mesh networks, multi-hop ad-hoc networks, etc. In this paper, we introduce a new approach for the forwarding and dropping dilemma that faces every node participating in a MANET. We assume that nodes in an ad hoc network behave as if they are in a competitive environment where each node seeks maximum delivery of its destined packets and minimum power consumption due to forwarding. The strategy makes use of data gathered from neighboring nodes that speaks about their behavior. Our strategy will eventually exclude selfish nodes and include cooperative nodes in accordance to the decision it makes.**

I. Introduction

Multi-hop ad-hoc networks are self-configuring networks that do not undergo certain paradigm [1]. Such type of networks' performance greatly depends on the nodes constructing the network. Nodes' behavior (such as: rate of mobility, packet dropping to forwarding ratio, etc), average number of hops and the average distance between nodes form an incomplete set of factors that influence the network's performance [2] [3] [4]. By definition, the natural structure of such wireless networks imposes a voltaic cell to be their power source. In addition, multi-hop ad-hoc networks host to host communication is only done by the forwarding support of other nodes (relay nodes [5]) that are not related to such session. As a result, each node faces a problem of ambiguity when being an intermediate station in the chain of hops linking two currently communicating end systems, and is asked to deliver a packet. Should the node participate in the delivery process of that packet or should it just drop it? An altruistic behavior of continuos forwarding causes a great deal of power consumption and should sometimes fail due to the intra-flow interference [5]. On the other hand, a selfish behavior of continuos dropping will eventually drive the node out of the network by being rejected from other nodes (as a sort of punishment).

Our proposed framework defines a new forwarding and dropping strategy that is neither considered as selfish nor altruistic. The strategy defines a decision engine that responds with a boolean answer of wether to drop or to forward a specific packet based on the packet's intended destination. Decisions made are tuned towards cooperation or defection on a node by node basis in accordance to a node's behavior and its general reputation. A decision is not only made based on a node's general behavior to all other nodes but to the way it behaves specifically to our node. That is to say, if the node is acting selfish to all other nodes but cooperative to our node, the decision engine will still behave altruistic to it.

The rest of this paper is organized as follows: in section 2, we discuss briefly some of the past related work done with the forwarding/dropping issue. Section 3 describes the proposed strategy in details and how it processes the data gathered from the surrounding nodes. At section 4, we present our simulation results that were tried on different sets of data. Finally, a brief conclusion summarizing the whole idea and how this work could be extended in the future is presented in section 5.

II. Related Work

Dropping event detection has been a point of interest for many researchers. Detection and mitigation of malicious packet dropping DoS attack can be done with a 76% accuracy at the worst case scenario using distributed probing technique [6].

Researchers have been motivated to write cooperation strategies that drop unrelated packets, while acting as relay nodes, due to multiple factors. Packet dropping may be motivated due to the lack of channel access opportunities on a partially congested route in a network [5]. In addition, dropping decision may be invoked for handling non-cooperating nodes as a sort of penalizing them and thus making it unattractive to deny cooperation [7]. Moreover, packet dropping may occur to lower class packets for the sake of assured forwarding of higher class packets while contention [8].

The MANIAC [9] (Mobile Ad-hoc Network Interoperability and Cooperation) challenge is a testing environment that was setup twice (2007 and 2009) for experimenting the interaction between different cooperation strategies in a MANET. The

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8379
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8379

interactions between multiple cooperation strategies that occurred at the 2007 MANIAC Challenge were discussed and the traffic across multi-hop nodes that were emulated from connectivity traces gathered was severely analyzed [10]. At the MANIAC 2007, the strategy winning award used ARP filtering and uni-casting of OLSR for connection minimization leading to reducing the node's power consumption [11]. Whereas at the MANIAC 2009, the strategy winning award presented an adaptive routing strategy based on a diversity paradigm which resorts to different node behaviors [12].

III. The Proposed Strategy

Basically, our proposed strategy depends on the activation of promiscuous mode to bypass sniffed packets, to the upper layers for investigating them. While achieving this, a couple of $(n+1)*(n+1)$ matrices, namely $droppedMat$ and $forwardedMat$, are to be built and stored at a node y, such that y is the node that implements the proposed strategy and n is the number of neighboring nodes. $droppedMat$ is, structured such that $droppedMat_{i,j}$ is the number of packets dropped by node i that were destined to node j. Similarly, $forwardedMat_{i,j}$ is the number of packets forwarded by node i that were destined to node j. Both matrices keep sensing the changes happening around (via the sniffed packets) and are to be updated according to these changes. All further processing will be done based on the data fetched from both matrices.

Firstly, lets start by defining the percentage of anger of node a from node b (i.e. how much angry is a from b) as:

$$anger(a,b) = \frac{droppedMat_{b,a}}{droppedMat_{b,a} + forwardedMat_{b,a}} \quad (1)$$

Based on such function, we calculate 2 factors namely $angrLvl$ and P_{rep}. $angrLvl_y$ is the percentage of anger of node y towards certain node in the network. The higher this value, the more angry y is from a certain node. Obviously, the more angry is y from a node, the more dropping to this node's packets should occur by y.

P_{rep} is the percentage of assurance that a certain node is a one that cares about reputation issues in making its forwarding/dropping decision. Caring about the reputation of a node means directing the tendency to either delivering or dropping its packets as a reflex for a cooperative or selfish behavior respectively. The higher this value is, the more forwarding should occur to that node for avoiding its punishment.

P_{rand} is a third factor that doesn't depend on the $anger$ function calculated in 1. It is defined as the percentage reflecting the assurance of a node playing a random or an undefined strategy that depends on a random factor (whether this factor is the sole decision making one or it shares some how in the decision making process). The more a node is found to be playing on a random basis, the more dropping to its packets should occur.

$angrLvl$ is a subjective percentage from the perspective of a certain node, however, the other 2 percentage values are general from the perspective of all other nodes in the network (i.e., don't change among all network nodes implementing the proposed strategy). All 3 factors are specific for a certain node and keep reconfiguring in relation to the data filling the prescribed matrices. Obviously, the more accurate the data, the better the obtained results.

A. Formula factors:

As mentioned, 3 factors are calculated out of the data matrices for executing the decision:

1) Anger Level: which is defined as

$$angrLvl_y(x) = anger(y,x) * 100 \quad (2)$$

and represents the percentage of anger of node y towards node x, such that $anger$ is function 1.

2) Randomness Probability: which is defined as a weighted moving average that links k_1 preceding decisions with the current one

$$P_{rand}(x)(m) = \begin{cases} 0.5 & 0 \le m < k_1 \\ a * P_{rand}(x)(m-1) & \text{case 1} \\ a * P_{rand}(x)(m-1) + b & \text{case 2} \end{cases} \quad (3)$$

Such that cases 1 and 2 are as:
Case 1: the last k_1 events are the same for c % of the nodes
Case 2: the last k_1 events are different for c % of the nodes
and represents the probability that a node is playing on an undeterministic basis at packet m. The constants a, b, c and k_1 are tunable for achieving the most precise detection. It is clear that if a node has dropped/forwarded k_1 consecutive packets to a certain node, it is intending to drop/froward to that node for certain reason, we are more sure it is not on a random basis.

3) Reputation Caring Probability: First, lets define $deflectionRatio$ as:

$$deflectionRatio(a,b) = \frac{min(anger(a,b), anger(b,a))}{max(anger(a,b), anger(b,a))} \quad (4)$$

Now we say:

$$P_{rep}(x) = \frac{\sum_{i=0, i\neq x}^{n} (-1)^y deflectionRatio(x,i)}{n-1} * 100 \quad (5)$$

and represents the percentage of a node x cares about others' reputation. y is such that:

$$y = \begin{cases} 0 & deflectionRatio(x,i) > k_2 \\ 1 & deflectionRatio(x,i) <= k_2 \end{cases} \quad (6)$$

where k_2 is a tunable constant for achieving the most precise detection.

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8379
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8379

B. Decision

In the final decision making process the $angrLvl_y$ and the P_{rep} are not directly plugged. Instead, they are applied to two factors in accordance to two different polynomial functions. As for the anger level, referring to the normal behavior of the human nature, it is considered wise to have a slowly reacting temper, i.e. to increase ones anger level slowly with respect to the surrounding actions. Nevertheless, at a certain instance, ones temper should eventually have rapid increase to stop some certain annoying actions as shown in figure 1(a). The $angerFact$, represented as the vertical axis in figure 1(a), is deduced from the $angrLvl$ in accordance to the shown relation. The shown curve is such that 50% $angrLvl_y$ reflects to 0.28 on the $angerFact$.

In the process of deducing the $repFact$, it is of a great importance to have an exponentially growing probability of a node cares about reputation. The reasons behind this are: 1) This factor has high influence on the node's score, i.e. once we get skeptical that a certain node might be caring about reputation, it is important to play safe with this node to avoid its anger. 2) The P_{rep} is the only factor in the final decision making function that is directly proportional with the decision making function, $P_{forwarding}$. Linear growth to that function will hold down the average value for $P_{forwarding}$ causing severe limitations to the total number of forwarded packets and eventual network exclusion. The $repFact$, represented as the vertical axis in figure 1(b), is deduced from the P_{rep} in accordance to the shown relation. The shown curve is such that 50% P_{rep} reflects to 0.8 on the $repFact$.

Meanwhile, at the case were all three factors ($angrLvl_y$, P_{rep} and P_{rand}) point to 0.5; $P_{forwarding}$ would have been 0.125 which is too low. After the $angrLvl_y$ and P_{rep} adjustments; $P_{forwarding} \approx 0.3$ at the same situation.

An online curve fitting engine [13] was used and the functions were obtained as follows:

1) Anger Factor:

$$angerFact(x) = A + Be^{-Dw} + Ce^{-Ew} \tag{7}$$

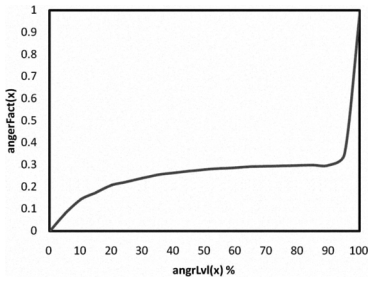
Such that:
w = $angrLvl_y(x)$ and A, B, C, D and E are tunable constants for achieving the optimum performance. The stated function yields the curve shown in figure 1(a)

2) Reputation Factor:

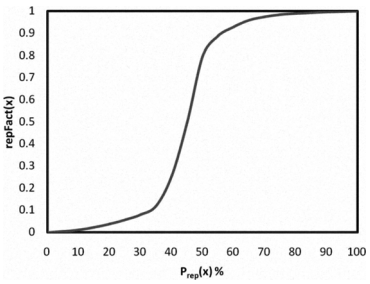
$$repFact(x) = \frac{A + Bz + Cz^2 + Dz^3}{1 + Ez + Fz^2 + Gz^3} \tag{8}$$

Such that:
z = $P_{rep}(x)$ and A, B, C, D, E, F and G are tunable constants for achieving the optimum performance. The function yields the curve shown in figure 1(b)

3) The final decision:
The final forwarding/dropping decision is made upon the receiving of a certain packet that is intended to node x and should be either dropped or delivered. We define the probability to forward a packet as:



(a) Graphical representation for function 7



(b) Graphical representation for function 8

Fig. 1. Anger and reputation curve mapping

$$P_{forwarding}(x) = (1 - angerFact(x))* \\ (1 - P_{rand}(x)) * repFact(x) \tag{9}$$

IV. Simulation

In this section, we provide the results obtained while simulating the proposed strategy in a multi-hop wireless ad-hoc network. Multiple runs were made each with different attributes considering: the number of nodes per network, the strategies implemented by each node and the average number of hops.

A. Assumptions

All runs were made with the following assumptions: The minimum capability of packet sniffing was adjusted at a 76% accuracy about all actions taken by most of the network nodes [6]. Packets are generated based on a random source to random destination according to a linear probability distribution function. Packet attributes (packet length, data contained, etc) are all kept constant.
In all the simulations, the randomness detection constants were adjusted as: k_1=4, c=60 %, a=0.85, b=0.15. Whereas the reputation detection constant k_2=0.9.

The score of node x upon receiving packet m was such that:

$$score_x(m) = \begin{cases} score_x(m-1) + 10 & m \text{ belongs to } x \\ score_x(m-1) - 1 & x \text{ forwards } m \end{cases} \tag{10}$$

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8379
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8379

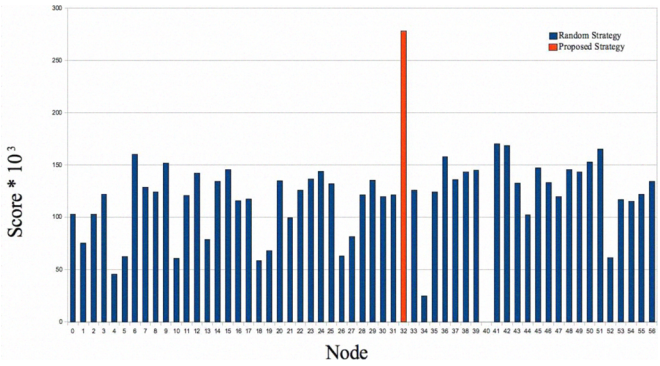


Fig. 2. Node's score calculated according to the function 10

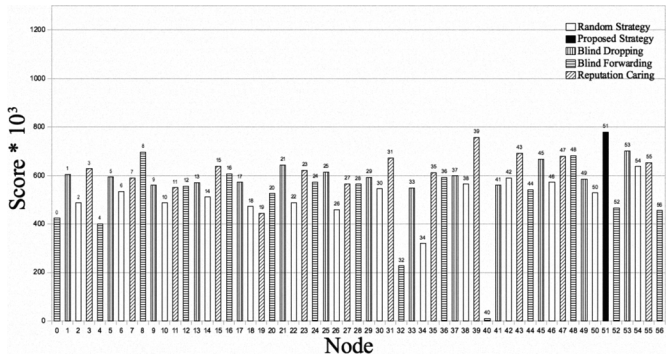


Fig. 3. Node's score calculated according to the function 10

B. Simulation Results

OMNeT++4.0 was used to perform all the following simulations. Multiple scenarios were made each with different strategy mixture. The first scenario was made on a network of 57 nodes with 56 of them playing a random strategy with a random threshold. All 56 nodes have been detected as playing randomly with a 100% accuracy. In this scenario, there were a total of 10 million generated and transmitted packets. Figure 2 shows the scores of each of the 57 nodes. The node indexed 32 (0-based) was the one implementing the proposed strategy and reaching the highest score: 278225.

The second scenario was made on the same network with the same number of nodes but with different strategy mixture. 13 nodes cared about reputation of others and used to punish uncooperative nodes according to their own anger level (calculated as in 2) towards each of the others. 14 nodes played random, 15 nodes were blindly forwarding packets and 14 were blindly dropping. Table I shows the reputation and randomness detection precision for each strategy category (reported as the average detection percentage for all nodes applying certain strategy). Figure 3 shows the scores of each of the 57 nodes. The node indexed 51 (0-based) was the one implementing the proposed strategy and, again, reaching the highest score: 25250.

TABLE I
AVERAGE *reputation caring* AND *randomness* DETECTION PERCENTAGE FOR EACH STRATEGY CATEGORY

Strategy Category	Reputation Caring (%)	Randomness (%)
Blind Dropping	10	0
Blind Forwarding	68	0
Reputation Caring	100	0
Randomness	7	100

V. Conclusion and Future Work

A new forwarding/dropping decision engine has been proposed in this paper. The proposed strategy depends on 3 main factors which are: the anger level, the randomness probability and the reputation caring probability. The anger level is how angry is a certain node from another, relative to the percentage of the dropped packets to the forwarded ones. The randomness probability is the probability of certain node playing a totally random or an undefined strategy that might or might not care about reputation, punishments or any other hypothetical factors. It depends solely on a random factor that shares directly or indirectly in the decision made. Last but not least, the reputation caring probability is the probability that a certain node cares about reputation issues and explicitly punishes a selfishly behaving node. These factors are obtained via the sniffing process done by a node for all neighboring nodes. All 3 factors share in the forwarding/dropping decision according to function 9. Simulation results proved a great deal of efficiency and the capability of a node, implementing the proposed strategy, in surviving in a multi-hop ad-hoc environment with the *relatively* minimum power consumption.

The proposed work could be extended by the inclusion of more factors like: the state of the routing table, the current power level, the length of the "single hop" list, the average degree of the network nodes, network density, congestion, throughput and average distance between nodes. The inclusion of all these factors might, theoretically, have a great deal of influence on the node's own score (i.e. power consumption).

References

[1] A. B. MacKenzie and L. A. DaSilva, *Game Theory for Wireless Engineers*. Morgan and Claypool, January 2006, vol. 1, no. 1. [Online]. Available: www.morganclaypool.com

[2] M. Gerla, L.-J. Chen, Y.-Z. Lee, B. Zhou, J. Chen, G. Yang, and S. Das, "Dealing with node mobility in ad hoc wireless network," *LECTURE NOTES IN COMPUTER SCIENCE*, pp. 69–106, 2005.

[3] V. Naumov and T. Gross, "Scalability of routing methods in ad hoc networks," *Performance Evaluation*, pp. 193–209, August 2005.

[4] L. Badia, N. Bui, M. Miozzo, M. Rossi, and M. Zorzi, "Mobility-aided routing in multi-hop heterogeneous networks with group mobility," in *Global Telecommunications Conference*, 2007.

[5] D. Jung and H. Lim, "Probabilistic packet dropping for load control in multihop wireless networks," in *Local Computer Networks, 2008. LCN 2008. 33rd IEEE Conference*, October 2008.

[6] M. Just, E. Kranakis, and T. Wan, "Resisting malicious packet dropping in wireless ad hoc networks," in *In Proc. of ADHOCNOW'03*. Springer Verlag, 2003, pp. 151–163.

[7] R. P. Bora, D. j Harihar, and S. Sehrawat, "Detection, penalization and handling of misbehavior in ad hoc wireless networks," in *IAENG International Journal of Computer Science*, 2007.

[8] S. Han, E. Ko, and I. Yeom, "Assured forwarding in multi-hop wireless networks," in *Mobile Ad Hoc and Sensor Systems, 2008. MASS 2008. 5th IEEE International Conference on*, October 2008.

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8379
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8379

[9] Wireless@VirginiaTech, "Maniac challenge." [Online]. Available: http://maniacchallenge.org/
[10] A. Hilal, J. N. Chattha, V. Srivastava, M. S. Thompson, A. B. MacKenzie, and L. A. DaSilva, "Interactions between cooperation strategies in mobile ad hoc networks," in *International Conference on Mobile Computing and Networking*, 2008, pp. 99–100.
[11] Klimek, V. Sidimak, and F. Jakab, "Mobile ad hoc networks interoperability and cooperation: the live and let live strategy," in *8th Scientific Conference for Young Researchers*, 2008.
[12] M. Caleffi, "Maniac challenge: A diversity adaptive approach for cooperative behavior," MANIAC, University degli Studi di Napoli Federico II - Napoli, ITALY, Tech. Rep., July 2009. [Online]. Available: http://maniacchallenge.org/caleffi_diversityapproach_abstract.pdf
[13] "Online curve fitting and surface fitting." [Online]. Available: http://www.zunzun.com/

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8379
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8379