2011 International Conference on Communication Engineering and Networks
IPCSIT vol.19 (2011) © (2011) IACSIT Press, Singapore

Improve Throughput of Ad-hoc Network by Used Trusted Node

Mehdi Safarpoor[1], Mehdi Dehghan Takht Foladi [2] and Nasim Bolandhemat[3]
[1,3]Science and Research Branch, Islamic Azad University, Fars, Iran
[2]Islamic Azad University , Qazvin Branch, Qazvin, Iran
e-mail:[1] safarpoor@fsriau.ac.ir, [2] dehghan@aut.ac.ir , [3] n.bolandhemat@gmail.com

Abstract— ad-hoc networks are formed without any kind of management, and mobile nodes send their packets through wireless communications. Each node in this type of networks can perform both routing and hosting. Maintaining this type of network requires cooperation between the nodes. Absence of cooperation and misbehaviors decrease efficiency and network throughput intensively. In this study a mechanism is introduced on the basis of financial credit and background of cooperation between neighboring nodes, in which each node while pushing forward the others' packets tries to direct the packet to an alternative route - with minimum cost - when encountering a misbehaving node, and meanwhile considers a penalty for that node. As all the nodes in the network show such a behavior, the misbehaving nodes are gradually either isolated or obliged to correct their behavior. This plan would eventually increase the number of forwarded packets resulting in enhancing network throughput; although it also results in boosting traffic load and end-to-end delay.

Keywords- ad-hoc network, misbehavior, credit, network throughput, selfish node.

1. Introduction

In the modern world, using computer networks has become an important aspect of human life, and a large number of day-to-day works are performed through computer networks, especially the internet. Therefore, use of such networks must be facilitated wherever the people live.

In ad-hoc networks, the network functions well only when all nodes participate in routing and forwarding the packets. Yet, due to limitations like battery life and low bandwidth, the nodes may not tend to cooperate in routing and forwarding others' packets; such nodes are called "selfish nodes".

This study aims to enhance network throughput and defeat selfish nodes, in a way that the selfish nodes are isolated, unable to use the network facilities, and unable to keep on selfishness.

Cooperation in ad-hoc networks, performing network tasks, including routing and forwarding others' packets, aims to help other nodes of the network. In this case, a node that disorders the routing and forwarding process and turns to a selfish node, may promote selfishness amongst other nodes of the network. Increasing number of selfish nodes would impose the entire network load on a limited number of nodes. Considering the increase of energy and bandwidth consumption under such a situation, the nodes would not be able to continue their cooperation for a long time, and the network would no more be able to function after a while due to inability of well-behaving nodes in routing and forwarding the packets.

Since the classical routing protocols like DSR [1], AODV [2], SMR [3], etc cannot effectively confront the misbehavior of middle nodes in the process of routing and forwarding the data packets, we need a mechanism that while serving each node, evaluates its level of cooperation in routing and forwarding the others' packets. Several mechanisms have been introduced in this regard, some of which are explained in brief in the next section.

2. Assessing the Existing Methods

2.1 Nuglet Algorithm

In [4] a virtual currency is introduced as “Nuglet”, which the functioning nodes may earn for the service they provide to the others, and can spend when sending their packets. This mechanism encourages the nodes to keep cooperating and keep their machines on in order to collect more Nuglets.

2.2 Span Algorithm

In this method [5] the nodes select the cooperating nodes on the basis of the information they receive from the network. In this algorithm, a node cooperates when it realizes that its neighbors cannot carry the packets, and otherwise it sets a bit - while routing - indicating that it cannot cooperate. Indeed, the cooperation mechanism in this method is an interval mechanism where a node cooperating in one interval rests in the next interval.

2.3 ODCCR Algorithm

In ODCCR[1] [6] a mechanism is suggested based on paying a cost to the cooperating nodes, where the first nodes finds a route with minimum cost, persuades the nodes on the route, and immediately gives them credit. When the destination node receives the packet, the credit is immediately exchanged for cash. In this method, if one of the nodes exhibits a misbehavior the forwarding process is continued by other middle-nodes and a “problem en-route” message (RERR) is transmitted to all nodes, so that the nodes above the misbehaving node do not receive next packets.

2.4 TMRP Algorithm

In TMRP[2] [7] unlike in previous methods which were based on single path routing, the authors believe that having several paths to the destination increases the system fault tolerance, and have developed their plan based on multiple path routing.

In this method, virtual payments against forwarding packets are used to encourage the cooperation, and it is mentioned that to enhance the cooperation we must withhold paying fix amounts to the cooperating nodes, and find the most economic way to the destination by tendering.

2.5 REEF Algorithm

In REEF[3] [8] method each node acts independently, not sharing any information with other nodes, and trusts only the information related to the nodes of the same age. In this method, each node not only doesn’t respond to forwarding, but also must direct the packets to a path with maximum success probability.

The proposed method is based on reliability of indicators. Each node dynamically updates its reliability table, including values or each neighboring output link. Each value is exclusive to each neighbor and shows reliability indicator of all paths passing through this neighbor (figure 2). Whenever a node sends a packet to a path, it updates values of its reliability table, and finds the values equal to those of all neighbors through which the packet has passed. If the packet is send successfully, the update is positive; if not it is negative.

3. Proposed Algorithm. Introduction

The proposed plan is a general plan that can be implemented in all routing protocols which are based on multiple path routing, and is capable of confronting misbehaving nodes and promoting cooperation in ad-hoc networks.

This plan is a combination of two methods based on credit and cost. Foundation of this plan is formed by the cost-based method, and the credit-based method is used to keep the record of nodes misbehavior.

Since choosing the most economic way requires tendering and assessing several offers, multiple path routing is the most important condition for implementing this plan.

---

[1] On-demand cost credit routing
[2] Truthful Multipath Routing Protocol
[3] Reliable and Efficient Forwarding in ad hoc networks

In this plan routing is completely separated from the forwarding system, and the system doesn't restrain participation of misbehaving nodes in the routing process. On the other hand, neighbors of the misbehaving node take its background into consideration while offering the cost of the path passing through that node, and try not to use the misbehaving nodes when forwarding the packets.

Here, each node has a routing table, which includes the next node, as well as cost of reaching the destination through that node. This would help calculate the total cost of reaching the destination when selecting the path, and prevent conspiracy of nodes in offering the cost. This would help preventing problems of protocols like Nuglet, due to which as the packet didn't have enough Nuglets, it was left in the midway.

This plan uses the cooperation remuneration of ACK.

3.1 Routing

Routing phase in the proposed plan is very much like that of AOMDV[4] [9]. Two tables of *multiple-path routing* and *neighboring* tables are added to each group in order to meet the objectives.

In this method, unlike in TMRP [8] where each neighbor offers its cooperation fee independent of the path, the offered price is the cost of sending the packet from the next node to the destination. This is very useful for selecting of optimum path (cost-wise) and preventing conspiracy of nodes when offering the price.

Figure 2 shows an example of routing table formed in node 21 of the network shown in figure 1 with 36 grid-like nodes. (In this network node 8 intends to send information to node 28)

0	1	2	3	4	5
6	7	8[s]	9	10	11
12	13	14	15	16	17
18	19	20	21	22	23
24	25	26	27	28[d]	29
30	31	32	33	34	35

Figure 1. Network by 36 nodes (grid)

destAddr= 8 *destSeq = 4* *hopCount = 5* *nextHop = 15* ***my_RouteCost=51*** *activated = 1* *lifetime:* *11424279406*	*destAddr= 28* *destSeq = 2* *hopCount = 2* *nextHop = 27* ***my_RouteCost=27*** *activated = 1* *lifetime:* *11463392056*	*destAddr= 8* *destSeq = 4* *hopCount = 5* *nextHop = 22* ***my_RouteCost=49*** *activated = 1* *lifetime:* *3494155711*
destAddr= 8 *destSeq = 4* *hopCount = 3* *nextHop = 20* ***my_RouteCost=35*** *activated = 1* *lifetime:* *3442373507*	*destAddr= 28* *destSeq = 2* *hopCount = 2* *nextHop = 22* ***my_RouteCost=18*** *activated = 1* *lifetime:* *11463392056*	*destAddr= 28* *destSeq = 2* *hopCount = 4* *nextHop = 15* ***my_RouteCost=39*** *activated = 1* *lifetime:* *21424279406*

Figure 2. Routing table formed in node 21

3.2 Selecting Next Path

[4] On-demand multipath distance-vector routing in ad hoc networks

The point of strength of the routing algorithm in this method – compared to the other methods – is the fact that, as shown if figure 2 – the cost of reaching the destination through each neighbor is known, and the packet holder can select the next node for sending the packet to the destination considering the lowest offered price.

The question is how we can ensure that middle nodes ask for the minimum enumeration for their cooperation!

Possibility of selecting a path from the existing paths, when we know how much it would cost to send a packet through that path, makes the middle nodes offer the best price in order to win the tender and collect the money (they spend this money on sending their packets respectively).Equations

3.3 Confronting Misbehaviors

In this plan, each node receives the minimum credit required for sending a packet 5 blocks a way as it joins the network. Since this credit is not enough to send all the packets, the node has to participate in sending the others' packets and earn some money. Therefore, this assumption causes no problems to the main idea of the plan, which is cooperation of nodes aiming to earn money for sending their packets. However, considering the fact that nodes are logic and they have limit resource, we may certainly say that nodes existing in a network see no reasons for keeping on cooperation after they get enough money for sending their packets throughout the network. On the other hand, the nodes would not do anything that may eventually cause problems to their trend of work in the network.

In this method, we use the principle, *the nodes would not do anything that may eventually cause problems to their trend of work in the network*, and consider punishments for nodes not cooperating in transmitting the others' nodes. This method requires neighboring nodes to know about the tendency of each node to cooperate with others, and its background of cooperation or misbehavior.

```
Neighbor Table{
 destAddr;
 number of  selefish behavior;                                        (1)
 number of  good behavior;}
```

This table is used when offering the price to each neighbor (Eq 2).

```
advertise Cost  = Route Cost  + ( cooperation_cost  *                (2)
number of selefish behavior );
```

3.4 Reacting to Misbehaving Nodes

As explained in previous section, the misbehaving node is punished by increasing costs and decreasing incomes; yet, we must think of a solution for the packet which has not been transmitted due to failure of the path. In our proposed plan, like in CHAMP method [10], each node has a small buffer for keeping transmitted packets, and when sending through a path fails, the system immediately tries another path (Fig 3). The important point in our proposed plan is that, unlike in CHAMP method, in our plan selecting the path is based on the cost, and middle nodes cannot replace the failed path with any paths, other than paths which can reduce transmission fee of the current node to make the total cost less than or equal to the previous path.

As shown in figure 3, misbehavior of node 7, causes node 5 to reduce its cooperation fee and select another path for sending the packet. Therefore, after selecting the support path and sending the packet through the second path, node 5 must consider a penalty for the misbehaving node (Eq 2).

3.5 Confronting Misbehaviors

In this section we qualify the plan and present the networking results. Simulation is performed by the software called "GloMoSim". The proposed plan is implemented on AODV routing protocol, and due to the necessity of implementing the algorithm on multiple path protocols, AODV algorithm is converted to

AOMDV algorithm, and then the proposed plan is implemented on it. Efficiency of the plan is assessed through various aspects and in various scenarios, and the simulation results are compared to AODV standard.

We continue by introducing criteria taken into consideration in efficiency assessment.

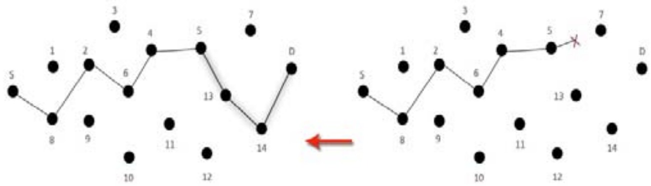

Figure 3. Changing route in node 5

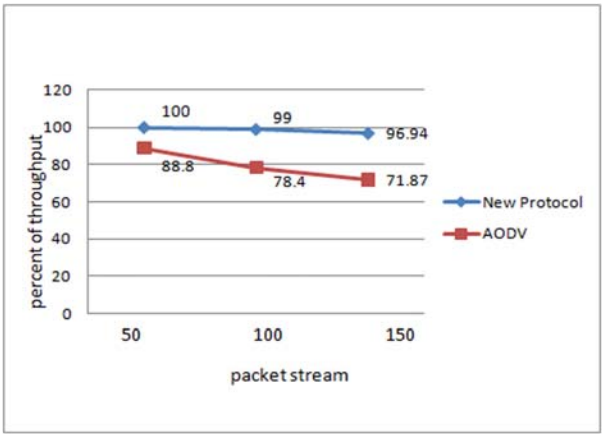

Figure 4. Network throughput with stream by 10 packet

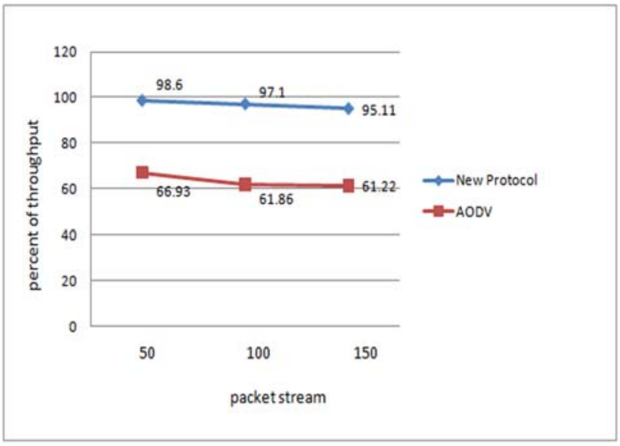

Figure 5. Network throughput with stream by 100 packet

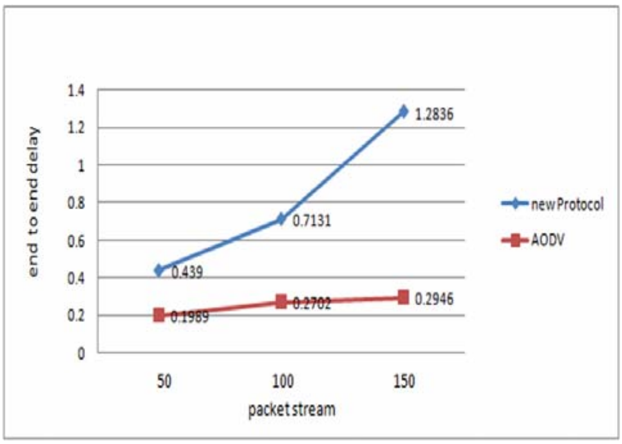

Figure 6. End to end delay

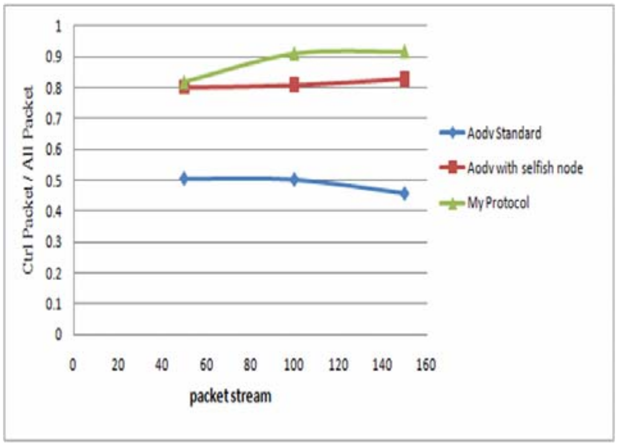

Figure 7. Traffic overhead in stream of packet

3.6 Throughput

As the plan aims to confront selfishness of nodes in order to enhance network throughput, throughput is considered the core criteria of qualifying the efficiency in this plan. (Eq 3)

DPR = Received Packets / Send Packets (3)

figure 4 shows throughput in a network with 36 nodes when the traffic load is 50, 100, and 150 stream of packets. As shown in the figure, increasing traffic load does not result in considerable fall of network throughput; while, sending the same data collection through AODV protocol, in which nodes act the rate of cooperation of 90%, decreases network throughput. In figure 5, by increasing the number of transmitted packet from 10 packets in each stream to 100 packets you can compare the effect of increasing network traffic on the proposed protocol with AODV protocol, where potential of cooperation is 90%. Comparing this figure to the figure 5 you'd realize that the number of forwarded packets in AODV protocol decreases considerably by increasing the network traffic, while increasing network traffic has a very small impact on efficiency of the proposed plan.

3.7 Traffic Overhead

End to end delay and traffic overhead in the proposed plan is shown by figure 6 and figure 7. Comparing it with traffic overhead of standard AODV protocol is indicative of increasing traffic overhead in the proposed plan. Increasing traffic overhead of the proposed plan is an outcome of using multiple-path protocols in this plan. However, as you see, using AODV protocol - with rate of cooperation of 90% - because increase number of RERR messages whenever the nodes misbehave and consequently increases traffic overhead intensively.

Analyzing Graph 3 shows that in AODV protocol the traffic overhead decreases by increasing number of streams. This is because middle nodes are aware of the path leading to the destination and consequently less broadcast packets are sended throughout the network. In the proposed plan, the middle nodes are not allowed to notify the path, because declaring the path requires to inform the cost, and the cost varies by the time. Traffic overhead of an AODV protocol with misbehaving nodes increases by increasing number of transmition packets; because, potential of misbehavior increases by increasing the number of packets passing through nodes

3.8 Conclusion

The proposed plan – compared to the plans explained in section 2 – can be more efficient because it uses a combination of cost-based and credit-based cooperation methods, and can be implemented on the classical routing protocols, which use routing tables. Potential of implementing this plan on the classical routing algorithm of ad-hoc networks makes it possible to use the classical algorithms in general ad-hoc networks without worrying about the problem of nodes cooperation.

Studying the simulation results shown in the graphs suggests that the proposed plan increases network throughput considerably. It also increasing the traffic overlhead; but the cost imposed on the network can be ignored as network throughput increases to 95%.

4. References

[1] David B. Johnson, Rice University "the dynamic source routing protocol for mobile ad hoc networks (dsr)" IETF manet working group 19 July 2004 .

[2] C. E. Perkins and E. M. Royer "ad hoc on demand distance vector (aodv) routing" draft-ietf-manet-aodv-02.txt, nov. 1998.

[3] S. Lee, M. Gerla, "split multi-path routing with maximally disjoint paths in ad hoc networks" proceedings of IEEE ICC 2001, vol. 10, June 2001, pp. 3201–3205.

[4] Levente Buttyan and Jean-Pierre Hubaux "Nuglets: a virtual currency to stimulate cooperation in self-organized mobile ad hoc networks " technical report DSC/2001/001, Swiss Federal Institute of Technology , 18 January 2001 .

[5] T. Dijkstra, H. Gautama and D. Epema "stimulating cooperation in ad-hoc networks" CACTUS Project in TU_Delft company 28 may 2004.

[6] Wesley Chee-Wah Tan *, Sanjay K. Bose "throughput and lifetime performance of cost–credit-based routing protocols for power constrained ad hoc networks" Computer Communications 31 , August 2008 , pp. 3964–3977.

[7] Yongwei Wang, Venkata C. Giruka, Mukesh Singhal "truthful multipath routing for ad hoc networks with selfish nodes" United States , March 2008.

[8] Marco Conti, Enrico Gregori, Gaia Maselli "reliable and efficient forwarding in ad hoc networks" Elsevier ad hoc networks 4, 2006, pp.398–415.

[9] M.K. Marina, S.R. Das, "on-demand multi-path distance-vector routing in ad hoc networks" proceedings of IEEE ICNP 2001, November 2001, pp. 14–23.

[10] Valera, W.K.G. Seah, S.V. Rao, "cooperative packet caching and shortest multi-path routing in mobile ad hoc networks" proceedings of IEEE infocom 2003, vol. 1, April 2003, pp. 260–269.