Improve TCP performance with Link-Aware Warning Method in Mobile Ad hoc Networks

Wenbo Zhu, Xinming Zhang, Nana Li
Department of Computer Science and Technology
University of Science and Technology of China
Hefei, 230027, P.R. China
xinming@ustc.edu.cn

Abstract—Along with the reactive routing algorithms of mobile ad hoc network (MANET), preemptive routing is a patch method which tries to warn the source node and to tell it to change route in time before a link is about to break. Preemptive method was proposed to decrease packet losses and improve TCP performance in MANET. However, the precision of warning occasion is an important problem, and affects TCP performance greatly. In this paper, with an improved link status evaluation technique, we present a LAW (Link-Aware Warning)-Routing method to improve the preemptive routing by increasing the precision of warning. We also choose an appropriate Reaction Time, which is an important factor in route warning occasion, according to our LAW-Routing function. The simulation result shows that the new method can improve the performance of routing and TCP greatly, especially in high-mobility scenarios.

Keywords—link status evaluation; preemptive routing; reaction time; route warning; TCP performance

I. INTRODUCTION

MANET is the network completely self-organizing and self-configuring, requiring no existing network infrastructure or administration. TCP [1] is a transport protocol designed to provide a reliable end-to-end delivery of data over unreliable networks, and it performs well over traditional wired network. However in MANET, TCP meets some severe problems [2], the primary one of which is that TCP is unable to distinguish the cause of packet losses between route failure and network congestion. Originally TCP assumes congestion to be the primary cause of packet losses and abnormal delays, but in MANET packet losses can be caused also by route failures. So when there is a dropped packet due to route failure, TCP will rashly believe that there exists congestion and do the relevant congestion avoidance methods, resulting in unnecessary shrink of congestion window and decline of throughput. If route failures happen frequently, the entire network will run in a very low throughput.

In order to alleviate the problem of TCP’s blindness to non-congestion-related packet losses, several schemes have been proposed [3]. Some methods (TCP-F [4], ELFN [2], ATCP [5]) try to tell TCP the true reason for packet losses through notification from routing layer. Fixed RTO [6] and TCP DOOR [7] try to find route failures through a simple measurement method upon RTO and the order of acknowledges.

The remainder of this paper is organized as follows. Section II introduces the related work. Section III describes the new method LAW-Routing. Section IV presents an experimental evaluation. Finally, Section V presents concluding remarks.

II. RELATED WORK

Preemptive Routing Method [8] is a patch on MANET routing protocols. Nodes in this method monitor the status of physical link and accordingly judge ahead that the route is about to break down, and then initiate a warning to notify the source to change route. The key to improve network performance of this method is the time of launching the warning. If warning too early, a route that still can be used will be replaced compulsorily. This will lead to unnecessary traffic from the route change. If warning too late, the source probably can’t find a new route timely before the former route breaks down. As a result, the preciseness of the warning mechanism is a crucial factor in the preemptive routing.

Unfortunately, the monitoring method upon links in [8] is not precise enough. The contention of MAC is an important factor, but the link failure caused by this (which could be called false link failure [10]) is temporary and recoverable. When two end nodes of a link move out of the transmission range of each other, the link will lose effectiveness in a long period or even permanently. Therefore, mobility is the most important factor on link status, and [8] [9] [11] and this paper consider this factor only. Generally, the information of mobility can be obtained with the help of the signal strength of received packets which can be gathered directly from physical layer.

\[ d = \frac{P_r \cdot G_s \cdot G_t \cdot R^2 \cdot H^2}{P_s \cdot L} \]  \hspace{1cm} (1)

Equation (1) [12] describes the fundamental of the two-ray ground propagation model, which is a generally used radio transmission model in simulations of MANET. With equation (1), we can calculate the distance \( d \) between nodes from the received signal strength \( P_r \), and combined with the factor of time, we can then get the velocity of nodes.

In [8], the mobility situation of all nodes is regarded equally and averagely and there could be a large number of
nodes with high speed that cannot change route timely! In contrast with this, [9]'s method of evaluating link status is more rational with a method called Proactive Link Management. In this method, each node keeps a table containing the distance information to its neighboring nodes. Concretely the table entry saves last time and this time the distances \(d_1\) and \(d_2\) from a certain neighbor (calculated from the signal strength of the received packet with equation (1)) and the timestamps \(t_1\) and \(t_2\) of packet receiving. As is shown in

\[
d_{hi} = d_2 + \frac{d_2 - d_1}{t_2 - t_1} \times (t + 0.1 - t_1)
\]

(2)

\(t\) is the current time, and from the equation we can get the distance \(d_{hi}\) from the certain neighbor. 0.1s (it can be regarded as the expected time cost of route change) later from \(t\). If \(d_{hi}\) is in excess of the transmission range of node, the current node could launch a warning (assume that the movement from \(t_1\) to \(t_2\) is linear).

[9] evaluates link status more reasonable, but in fact [9]'s method needn’t to consider about the preciseness of warning. This is because in [9] there is another method called Reactive Link Management which makes the current node to increase the transmission power in order to re-establish a broken link temporarily. So ideally, for this mechanism [9] can guarantee no packet losses from warning failure. However, the behavior of increasing nodes’ transmission energy is very harmful to the global network, as it will cause the decrease of spatial reuse and intensify link contention.

Both [8] and [9] have their own advantage, and if [8]'s preemptive routing method and [9]'s evaluation method of link status can be combined together rationally, TCP performance could be improved accordingly. In order to increase the precision of preemptive warning, we proposed a routing method called LAW (Link Aware Warning)-Routing, which can cooperate with different routing algorithms, such as DSR [13] and AODV [14], as a routing patch. To make a compare with preemptive routing on DSR (PDSR) of [8], we implement our LAW-Routing on DSR too.

III. LINK-AWARE WARNING (LAW)-ROUTING

LAW-Routing improves the preemptive routing method on warning precision with a new link status evaluation method. With this method, link status can be evaluated more accurately. Based on the technique of [11], it also solves the problem of disabled route cache in preemptive routing. The detail of LAW-Routing will be introduced as below.

A. Link Status Evaluation

In our LAW-Routing, the method of link status evaluation is similar to [9]. In this method, wireless physical agent will get the signal strength of a received packet, and save it in the packet.

First of all, each node keeps a hash table that records information of its neighbors. An entry in the table includes a neighbor ID \(id\) (the key in hash table), a signal strength \(p_1\) and a timestamp \(t_1\) of the last received packet, a signal strength \(p_2\) and a timestamp \(t_2\) of the received packet being handled at present, a velocity \(v\) between the current node and the neighbor, and an energy warning threshold \(th\). Every time the current node receives a packet from a specific neighbor, it will fetch the corresponding entry in the table and record the signal strength and the current timestamp in it. Subsequently from \(p_1\) and \(p_2\), we can calculate the distance from the current node to its neighbor with equation (1), and we name them \(d_1\) and \(d_2\).

\[
v = \frac{d_2 - d_1}{t_2 - t_1}
\]

(3)

With equation (3), we can obtain the velocity between the two nodes, and update \(v\) with it. If \(p_2 < p_1\), which means that the two nodes are moving farther from each other, we will calculate and update the energy warning threshold \(th\) with the equation below (the detail of deduction process of this equation can be found in [8]).

\[
\text{th} = \text{Prange} \times \frac{\text{range} - v \times \text{RT}}{\text{range} - v}
\]

(4)

In equation (4), \(\text{Prange}\) is the minimum power receivable by the device of nodes, \(\text{range}\) is the transmission radius, \(\text{RT}\) is the time span used for route warning process of routing agent [11].

Finally, the agent will check if \(p_2\) is less than \(th\), and if so, it means that the link between the two nodes will break down. We add a flag \(\text{is\_warn}\) in the packet, and at this time the current node will open \(\text{is\_warn}\) in this packet and send it up to routing agent.

Considering the case that \(p_2\) is greater than \(p_1\), we can find that the update of \(v\) is not necessary, because in this case there won’t be a warning launching. However, we should consider the fluctuation of wireless channel, so the relationship between received energy and distance of two nodes does not depend on equation (1) strictly. As a result, the more information we utilize from physical layer, the more precision we can get to calculate velocity between two nodes. On the other hand, we use equation,

\[
v = \frac{1}{4} v_{\text{new}} + \frac{3}{4} v_{\text{old}}
\]

(5)

in implementation, and try to get more precision with this smoothing strategy. Considering the variation of node speed is not linear and continuous, we put more weight on the newly obtained velocity \(v_{\text{new}}\) in contrast with the former velocity \(v_{\text{old}}\) in the entry.

B. Routing Method

Based on the work of [11], we design a routing patch method Link Aware Warning (LAW) routing instead of preemptive routing. The initiation of route warning is packet-driven, that is a node will find the instability of a link from \(\text{is\_warn}\) flag in packets. However besides this fact other conditions should be also satisfied: (i) This packet should not be received by snooping. (ii) This received packet should be a data packet, because a control packet is always produced by a temporary source node which will not send any subsequent packets after this control job. (iii) The source of this warning target is not being warned at present, and we can check this with a so called \(\text{src\_warning}\) array, which contains entries of sources that are being warned at present. When the current node decides to launch a route warning, it will initiate a Route Warning control packet (RWRN), write the information of the
warning link in it, and send it to the source with the reversed route of the received packet.

When the source node receives a RWRN, it will initiate a route change towards the destination node announced in RWRN. Firstly the node will refresh the cache, and initiate a route discovery if no according cache route remaining. The Route Request (RREQ) will record the information of the warning link, and the intermediate nodes that receive this RREQ will refresh their route cache according to the bad link information in the packet. Additionally a replied route that contains a warning link will be useless, so intermediate nodes will only forward the RREQ with received signal strength larger than warning threshold.

After the source finishes the route change, a Route Error (RERR) message will be sent to the warning node about the warning link. This RERR will ensure the refreshment of cache at the intermediate nodes from the source to the warning node. It is necessary because these intermediate nodes are still using routes with that bad link. Finally when the warning node receives the RERR, it will also remove the entry of this source node from src_warning array. Then the warning node will not believe that this source is still being warned afterwards. At this time, the whole warning process is accomplished.

C. Reaction time

In LAW-Routing, reaction time is an important factor which determines how long before the link breakage to launch a route warning, and this time span is used for route switch at routing layer. The definition of reaction time [11] is as follow,

\[ RT = 2 \times T_{\text{hop}} \times N_{\text{hop}} + T_{\text{route\_change}} \]  

(6)

where \( T_{\text{hop}} \) is the delay per hop which means the time for a packet transmitted from a node to another neighbor node, \( N_{\text{hop}} \) is the number of hops between the warning node and the source node, and \( T_{\text{route\_change}} \) is the time for a route change.

IV. EXPERIMENTAL STUDY

We studied original DSR [13], Preemptive DSR (PDSR) [8], and LAW-Routing on DSR with NS-2 network simulation tool [12]. The scenario contained 50 mobile nodes, which moved according to random waypoint model in a rectangular area of 1000 by 1000 meters. In this mobility model, a node randomly picks a location within the simulation area and starts moving straight towards it with a random constant speed ranging in \((0, \text{maxspeed})\). Every time the node reaches the destination, it will pause for a period of time (0s in simulation), before another new movement with random parameter values but same style. We set 50 TCP connections between random selected sources and destinations and the traffic over them was an FTP application with infinite length data transfer. The simulation time was 500s, and all the connections were established at a random time from 0s to 100s.

In order to calculate the received energy warning threshold \( th \), some parameters should be given here. As is shown in equation (4), the transmission range \( range \) was set to be 250m, the minimum power receivable by the device \( p_{\text{range}} \) was set to be 3.658e-10j. We also had to get the value of reaction time \( RT \) and we sampled the three parameters \( T_{\text{hop}}, N_{\text{hop}}, T_{\text{route\_change}} \) in equation (6). For this, we ran the network on original DSR with the same scenario and traffic style as the performance research simulation. Specifically, we generated ten scenarios containing a set of 50 nodes with maximum speed 15m/s and finally we obtained the average result that \( T_{\text{hop}}=0.05s, N_{\text{hop}}=2.229508s, \) and \( T_{\text{route\_change}}=0.253627s \), so according to (6), we can get the value of \( RT=0.476588s \).

Because mobility plays an important part in the precision of link status evaluation, we mainly focused on the performance upon different node speed in our experiments. In a specific scenario, the speed of nodes were selected in \((0, \text{maxspeed})\), and we set \( \text{maxspeed} \) to be 5.0, 7.5, 10.0, 12.5, 15.0, 17.5, 20.0(m/s) in different experiments. In a definite value of speed, we randomly set up 10 different scenarios as experiment samples, which meant that each result data shown in figures was the expectation of 10 direct simulation result data.

![Figure 1. TCP overhead](image)

Fig. 1 shows the overhead of control packets under TCP of the three methods, which is evaluated by counting the total number of routing control packets in the network. As is shown in the figure, LAW-Routing gains 8% and 70% decrease on the overhead in contrast with original DSR and preemptive DSR. It is because of the reservation of route cache that we can obtain such great improvement to preemptive DSR. In theory, LAW-Routing cannot gain too much improvement to original DSR, because though this method decreases packet losses to a large extent, it still has to inject all the route error messages into the network in order to reserve route cache. However we still acquire some improvement on overhead to DSR, and the reason could be that original DSR may get a new route containing an instable link while route discovery, but in LAW-Routing there is no such case.

Fig. 2 shows the throughput under TCP of the three methods, which is evaluated by counting the total number of data packets that are received by destination nodes. Throughput is an important factor for TCP [2] because of TCP’s blindness to non-congestion-related packet losses. LAW-Routing gains 20% and 9% increase on the throughput in contrast with original DSR and PDSR, and we can see that preemptive method can improve TCP throughput greatly. LAW-Routing improves PDSR because of the modification of link status evaluation method, which increases the precision of warning and decreases the number of warning failures greatly. What
should be noticed is that the performance of the new method is not affected by speed at all, profiting from the dynamic and precise monitoring of node speed.

![Figure 2. TCP throughput](image2)

![Figure 3. TCP end-to-end delay](image3)

Fig. 3 shows the end-to-end delay under TCP of the three methods, and this is another important performance factor of TCP. LAW-Routing gains 18% and 10% decrease on the end-to-end delay in contrast with original DSR and preemptive DSR. There are mainly two reasons for the decrease to original DSR. One is that in the course of warning, route switch and packet transfer with the former route can proceed simultaneously at source, and the other is the timeout and retransmission of packet losses after route failures. There're also two reasons for the decrease to PDSR. One is the reservation of cache and the other is the timeout and retransmission of packets losses after warning failures.

From these three figures we can see, velocity plays an important role in the performance. Along with the speed increases, network topology changes constantly, life of links decreases and route continually changes. As a result, route failures happen frequently, and TCP performance decreases to a large extent because of packet losses. PDSR obtains some improvement on network performance, but because of the equalization of node speed in PDSR, the precision of link status evaluation is too low. Especially in high-mobility scenarios, PDSR performance decreases to some extent. However in contrast with PDSR, velocity does not have any influence on LAW-routing profiting from an improved link status evaluation method. Adopting the smoothing strategy of speed information updating per packet, physical layer can measure and calculate the speed to neighbors accurately and this guarantees the effectiveness of warning launching in high-mobility scenarios.

V. CONCLUDING REMARKS

In order to alleviate the blindness of TCP to non-congestion-related packet losses, preemptive routing tries to advance the stability of route by changing route ahead before it breaks down. This method greatly depends on when to tell the source to change route ahead, and the performance will be degraded if warning too soon or too late. We presented a method called LAW-Routing to solve the precision problem of preemptive routing. It utilizes an improved link status evaluation method with regard to node mobility. In addition at routing layer, we reserve the route cache which is disabled in preemptive routing. We also choose an appropriate definition of reaction time, and with a rational value of reaction time, LAW-Routing could gain more precision at warning. Finally the simulation result manifested that TCP performance was improved greatly in LAW-Routing.

REFERENCES