Bloom Filter Based Load Balancing Mechanis for Mobile Ad Hoc Networks

Peiyun Luo
Electronic Material Evidence Forensic Center, Chongqing Municipal Public Security Bureau, China
Email: luopy2016@sina.com

Abstract —Load balancing mechanism in Mobile Ad hoc Networks (MANETs) can distribute the traffic evenly into the network and partial congestion can be alleviated. In this paper, a load balancing mechanism is proposed. The Bloom filter is used to detect the current flows in each node, which is a simple space-efficient randomized data structure for representing a set in order to support membership queries. The estimated link lifetime, the end-to-end delay and the existing traffic along the forwarding paths are used together as the routing metric. Our proposed load balancing mechanism is evaluated through simulation, and the performance is better than the traditional routing mechanism, such as the packet delivery ratio and the end-to-end delay.

Index Terms—Mobile ad hoc networks, load balancing, flow detection, link lifetime estimation, bloom filter

I. Introduction

MANETs are the multi-hop self-organized network without fixed infrastructure. The main purpose of Ad hoc technology is to set up temporary network with a group of mobile terminals in the absence of base stations [1]-[2].

For the wireless medium has the feature of sharing, the network resources are contented seriously [3]. The network performance can be enhanced through the well designed load balancing mechanism. By separate the flows into a proper size and sent along multiple routes, network congestion can be alleviated and the collision probability of the frames can be reduced. In order to provide accurate judgments for load balancing, flow detection mechanism is essential to determine the current traffic of the intermediate nodes.

Many researches [4]-[5] have shown that load balancing of the network can be achieved by using multiple paths to forward messages between any source-destination pair of nodes. The common belief that the same is true for MANET, multi-path routing balances the load significantly better than single-path routing. However, some researches [6]-[7] show that this is not necessarily the case.

There are several criteria for comparing single-path routing and multi-path routing in MANET. First, the frequency of route discovery is much less in a network which uses multi-path routing, since the system can still operate even if one or several end to end paths fail. Second, since multi-path routing distributes the load better, the overall throughput would be higher. But the nodes along the multiple paths may be close to each other geographically, due to the sharing character of the wireless channel, the nodes cannot transmit or receive simultaneously, it is said to be the inter-flow interference caused by route coupling [8]. Thus the performance of multi-path load balancing mechanism should be evaluated under specific scenario [9].

Multi-path and single-path routing mechanisms suit for different scenarios, for example, if the node density of the network is sparse, the path broken probability caused by node mobility should be the greatest contributor to the degradation of network performance. Since the end-to-end path has to be re-established frequently, the traffic is blocked since there is no route to the destination. This paper aims to evaluate the performance of multi-path load balancing mechanism in the sparse scenario, which is similar as the practical scenario.

In this paper, we propose a novel load balancing mechanism based on the Bloom filter [10], which is used to detect traffic load in each node. With storage space efficient manner, the traffic conditions of the intermediate nodes can be collected by the source node. Thus, the light traffic path can be selected to forward traffic.

Estimated link residual lifetime can be used as a metric to qualify the paths, for the path with the longer lifetime can lasts longer than the path using the hop count metric; furthermore, the number of re-routings can be reduced.

The contributions of this paper are illustrated below:

1) A scalable flow detection mechanism based on the usage of Bloom filter is introduced, where the traffic load of node is represented by the number of flows it carrying;

2) An efficient link lifetime estimation mechanism is introduced, this mechanism can be applied under different radio propagation models and mobility patterns; additionally, this mechanism does not require the aid of the positioning system;

3) For the sparse scenario, load balancing mechanism based on the results of flow detection and the estimated path lifetime is introduced, the performance of our mechanism is evaluated.

The rest of this paper is organized as follows: in the next section, we summarize the related works. In Section

Manuscript received June 24, 2016; revised November 21, 2016.
Corresponding: Peiyun Luo, email: luopy2016@sina.com.
doi:10.12720/jcm.11.11.1012-1019

III, Bloom filter based flow detection mechanism is introduced. In Section IV we introduce a link lifetime estimation mechanism and propose a load balancing mechanism. Performance evaluation and analysis are shown in Section V. Finally, conclusions and our future work are presented in Section VI.

II. RELATED WORK

As soon as Mobile Ad hoc Networks (MANETs) were proposed, researchers started to envision using end to end communications for content delivery. Pelusi *et al*. [11] studied opportunistic networking and discussed possible scenarios of its use. In the context of using smart devices with limited energy for content delivery, Li *et al*. [12] formulated the optimization problem of opportunistic forwarding, with a constraint concerning the energy consumed by message delivery for both two-hop and epidemic forwarding.

Authors in Reference [13] propose a dynamic load-aware reactive routing protocol for MANET. In this scheme, each intermediate node counts the pending packets in its outgoing buffer and accumulates this information to the re-broadcasted Route Request (RREQ) message. The destination node compares the arrived RREQ packets from different routes and selects the one with minimum path load. However, this scheme only counts the outgoing buffer size at each node, and does not consider potential traffic, thus the outgoing buffer size cannot reflect the actual network load accurately.

Based on the load status of each nodes, the RREQ messages are dropped actively in Reference [14]; moreover, the work load based adaptive load balancing technique is proposed. RREQ messages are forwarded selectively according to the load status of each node, so that overloaded nodes can be excluded from the requested paths. Each node begins to carry additional traffic flows again whenever its overloaded status is dissolved.

Reference [15] proposes a prediction based adaptive load balancing mechanism for MANET. This mechanism can operate with any kind of multi-path source routing protocol. Based on the measurement and prediction of network traffic, the traffic loads among multiple disjoint paths are distributed effectively. In order to predict network traffic accurately, the MANET traffic is recorded in their testbed.

Reference [16] proposes a link-disjoint loop-free multi-path routing mechanism named Trust Based Secure on Demand Routing Protocol for MANETs (TSDRP), which is the multi-path version of Ad hoc On-demand Distance Vector (AODV) [17].

However, the MANET has the feature of dynamical and sharing, different amount of network resources may be consumed by the same service under various scenarios. Therefore, the traffic load predicting is too difficult and important. In previous studies, the traffic load predicting is not considered.

III. BLOOM FILTER BASED FLOW DETECTION MECHANISM

Different with the definition in traditional wired network, the representations of network load in MANET are quite difficult. Firstly, the contention process should be executed before the data packet is transmitted. For the packets with different length, the propagation times are quite various, but the amounts of network resources consumed for medium access are equal nearly. Secondly, the current traffic load can be represented by the parameter of packets number processed by the nodes. But how many packets should be processed in the future can not be estimated accurately. Therefore, the status of resource contention is distinct obviously for the case of multiple packets from the unique flow and the single packet from several flows. For example, there are M flows in the network, and the flow can be represented by F_i, $1<i<M$, for the first case, K packets are all from the single flow F_i, and for the latter case, all the packets are from different flows. According to the analysis, the network load of the latter case may be heavier than the first case for the duration of end to end service is random. In our flow detection mechanism, the number of flows carried by the nodes is utilized to reflect the traffic condition of specify node, and a Bloom filter based flow detection mechanism is introduced in this section.

The Bloom filter is a simple space-efficient randomized data structure for representing a set in order to support membership queries. Bloom filters allow false positives but the space savings often outweigh this drawback when the probability of an error is made sufficiently low. Burton Bloom introduced Bloom filters in the 1970s [10], and ever since they have been very popular in database applications. Recently they started receiving more widespread attention in the networking application.

A Bloom filter for representing a set $S = \{x_1, x_2, \dots , x_n\}$ of n elements is described by an array of m bits, initially all set to 0. A Bloom filter uses k independent hash functions $h_1,\dots, h_k$ with range $\{1,\dots, m\}$. For mathematical convenience, we make the natural assumption that these hash functions map each item in the universe to a random number uniform over the range $\{1,2, \dots , m\}$. For each element $x \in S$, the bits $h_i(x)$ are set to 1 for $1\leq i\leq k$. A location can be set to 1 multiple times, but only the first change has effect. To check if an item y is in S, we check whether all $h_i(y)$ are set to 1. If not, then clearly y is not a member of S. If all $h_i(y)$ are set to 1, we assume that y is in S, although we are wrong with some probability. Hence, a Bloom filter may yield a false positive, where it suggests that an element x is in S even though it is not.

Fig. 1 provides an example. The filter begins as an array of all zeroes. Each item in the set x_i is hashed k times, with each hash yielding a bit location; these bits are set to 1. To check if an element y is in the set, hash it k times and check the corresponding bits. The element y_1 cannot be in the set, since a 0 is found at one of the bits.

The element y_2 is either in the set or the filter has yielded a false positive.

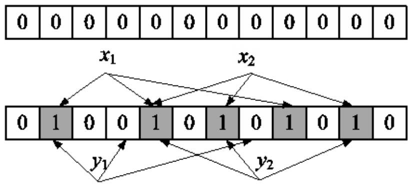

Fig. 1. An example of a standard bloom filter

For many applications, false positives may be acceptable as long as their probability is sufficiently small. To avoid trivialities, we will silently assume from now on that $kn < m$.

Given the assumption that hash functions are perfectly random, the probability of a false positive for an element is not in the set, or the false positive rate, can be estimated in a straightforward fashion.

After all the elements of S are hashed into the Bloom filter, the probability that a specific bit is still 0 can be expressed as Eq. (1):

$$p = \left(1 - \frac{1}{m}\right)^{kn} \approx e^{-kn/m} \tag{1}$$

Thus the probability of a false positive is given:

$$f = \left(1 - \left(1 - \frac{1}{m}\right)^{kn}\right)^{k} \approx \left(1 - e^{-kn/m}\right)^{k} \tag{2}$$

Bloom filter can be used here to simplify the flow detection process and reduce the storage space. Each node has a standard Bloom filter to record the flows carried by the node. Since the amount of the control packets is rather less than that of data packets, therefore only the data packets are recorded and the calculation is done for each incoming data packets. Each flow from the same source node is added with flow identification to distinct different flows between the same source and destination pair.

The source address, the destination address and the flow identification (*SrcDestFlow*) are used to identify different flows at a node. If we use integer counter to count amount of the flows, a hash table indexed by *SrcDestFlow* must be established for classifying each incoming packet, thus the storage space will be growing with the number of different flows. However, the storage cost is constant when the Bloom filter is used to identity flows.

The node checks whether the packet is recorded by indexing its *SrcDestFlow* value in the local standard Bloom filter whenever data packet is received successfully. If the flow is recorded, it needs not to be processed; otherwise, the *SrcDestFlow* value of the packet should be added in the Bloom filter.

The standard Bloom filter cannot delete an expired entry if a flow is finished; furthermore, tt is also complicated to record the starting and ending time of the flows. In order to acquire the present condition of flows in each node, the local standard Bloom filter will be reset periodically, the number of flows between the update intervals is calculated and stored in a local variable N_{flow} by (3) and (4). Therefore, when the parameter of N_{flow} is used by the load balancing process, it will be added to the corresponding field in the Route Reply (RREP) message.

In standard Bloom filter, we have the probability of the 0s in (1). The number of 0 bits for a set S is strongly concentrated around its expression $m(1-1/m)^{kn}$ in (3). So the relationship between the number of zeros and the number of elements in the set is shown as (4), where n is the number of the flows carried by the node.

$$P_{zero} = m\left(1 - \frac{1}{m}\right)^{kn} \tag{3}$$

$$n = \frac{\ln\left(P_{zero}/m\right)}{k\ln\left(1-1/m\right)} \tag{4}$$

IV. Multi-Path Load Balancing Mechanism

The load balancing mechanisms are realized with the route discovery process, after the routing mechanisms choose the proper path for the traffic according to the traffic condition of the intermediate nodes and the estimated link lifetime of the forwarding path. As a result, the traffic load can be distributed evenly into the network.

A. *Effective Link Lifetime Estimation Mechanism*

Our link lifetime estimation mechanism aims to predict the residual link lifetime which is defined as the remaining time that the two nodes within the transmission ranges of each other. Each node's movement consists of a sequence of random time intervals called mobility epochs, during which the node moves towards to a constant direction at a constant speed.

In our mechanism, the values of received signal strength are used to calculate the distance between the nodes, and they are transmitted from the radio receiver to the network layer.

For different radio propagation models, such as the Free Space Propagation model and the Two Ray Ground Reflection Propagation model, the received signal strength has the direct relationship with the distance between the sender and the receiver. Thus our mechanism can be applied under different radio propagation models.

The calculation process of the link lifetime between two nodes is illustrated below. The transmitting power of each node is supposed to be the same, and the received signal strength is determined by the distance between nodes. The omni-directional antenna is used, and P_t is the transmission power used for the signal transmission, then the receiving power P_r is proportional to $1/d^2$, where d is the distance between sender and receiver in open space, which can be expressed as (5). G_t, G_r, λ and L are constant in our calculation.

$$P_r = \frac{P_t G_t G_r \lambda^2}{(4\pi d^2)L} \tag{5}$$

The process of relative movement process and link lifetime estimation is shown in Fig. 2, where *TR* is the node's transmission range. Node S is the calculating node, the lifetime of its neighbour links are calculated and stored in node S. The current neighbour node position is point D, and point A and C are the positions of the same neighbour node while it sending messages to node S, thus only two stored values of the received signal strength are required for each neighbour, which reduces the storage and the calculation overhead dramatically. As can be seen that point B is the position that the neighbour moves out of the transmission range of node S. Obviously, the link lifetime is determined by the distance *L* and the relative node speed *v*. The received signal strength values are collected when the estimating node receives the packets from its neighbours. As Fig. 2 shown, Δt_1 and Δt_2 are defined as the time interval between the latest three received signal strength values.

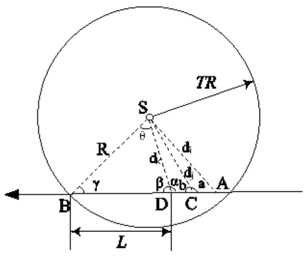

Fig. 2. Link lifetime estimation process

The cosine value of $\angle a$ and $\angle b$ can be obtained according to the law of cosines, for $\angle a + \angle b = 180°$, that is $\cos a = -\cos b$, we have:

$$\frac{(v\Delta t_1)^2 + d_j^2 - d_k^2}{2d_j v\Delta t_1} = -\frac{(v\Delta t_2)^2 + d_j^2 - d_i^2}{2d_j v\Delta t_2} \tag{6}$$

where the distance between the calculating node and the positions of the same neighbour node at different time, d_i, d_j and d_k, can be calculated from the received signal strength value, so the relative moving speed can be obtained:

$$v = \sqrt{\frac{d_i^2\Delta t_1 + d_k^2\Delta t_2 - d_j^2(\Delta t_1 + \Delta t_2)}{(\Delta t_1 + \Delta t_2)\Delta t_1 \Delta t_2}} \tag{7}$$

The calculated relative speed value from (7) is stored locally, and the stored speed values are approximately taken as an integer; furthermore, the periodically broadcasted Hello message of the on-demand routing protocol can be used to obtain the sampled receiving power, thus there are no extra control messages. The intervals between the Hello messages are not equal, since these intervals are jittered by the routing mechanism to avoid collisions. The received sample power values are stored periodically in the calculating node for the estimation of link lifetime.

To estimate link lifetime between the nodes, the procedure of computing L is shown below. According to the law of sines, in ΔSBD, the following equations can be obtained:

$$\frac{R}{\sin\beta} = \frac{SD}{\sin\gamma} = \frac{L}{\sin\theta} \tag{8}$$

$$\angle\gamma = \sin^{-1}\frac{SD\sin\beta}{R} \tag{9}$$

The three sides in ΔSAD are already known, and $\angle\beta = 180° - \angle\alpha$, so

$$\angle\alpha = \cos^{-1}\frac{SD^2 + AD^2 - SA^2}{2\times SD\times AD} \tag{10}$$

For $\angle\theta = \angle\alpha - \angle\gamma$, the expression of *L* is

$$L = \frac{R\sin\theta}{\sin\beta} \tag{11}$$

Therefore, link lifetime can be expressed as (12):

$$T_d = \frac{R\sin\theta}{v\sin\beta} \tag{12}$$

As introduced above, without the aid of positioning devices, our link lifetime estimation mechanism is much easier to realize. Our mechanism needs not to predict the speed and the moving direction of the node, it just estimates the link lifetime counted from the current calculation period. According to the mobility models, the moving direction and the speed change after each mobility epoch, therefore the calculation result may not be correct if the moving states of node change during the calculation period, which means the three received signal power values are not collected during the same mobility epoch. However, only the change of the moving speed is needed to be detected. After detecting the change of moving states, the calculation node will remove the useless received power values, and record the received power values from then on. The estimated link lifetime value will be set to Invalid at this calculating process, and it will not be used by the route establishing process at this time. Since the estimated results can only reflect the condition that the moving characters are unchanged during the calculation procedure, so the calculating interval is important for improving the accuracy of our link lifetime estimation mechanism. The estimation errors were illustrated in Section V.

The overhead of our link lifetime estimation mechanism is the extra storage space for the received and calculated values. Three signal strength values, one estimated current speed value and one estimated link lifetime value are stored in a structure of a hash table which is indexed by the address of the neighbour node. Each time the node receives the Hello packets from its neighbours, the signal strength value will be restored in the hash table, and the link lifetime estimation process will be triggered. Once the estimated node speed and link lifetime have been calculated, they will be restored in the table for the detecting the speed changes and the route establishing process.

B. End-to-End path Establishing and Load Balancing Process

The network traffic load can be balanced by forwarding the packets through multiple node-disjoint paths. But the number of control packets increase dramatically by excessive using the end-to-end forwarding paths. To trade-off the overhead and load balancing, the maximum forwarding path number for our load balancing mechanism is restrict to two, and three parameters are used for the forwarding path establishment, the end-to-end delay, the estimated link lifetime and the node traffic conditions of the intermediate nodes respectively.

There are three important parameters: 1) the maximum traffic of all the nodes along each reverse path $\max(N_{i,j} \times B_{i,j})$, where $N_{i,j}$ ($1 \le i \le 5$, $1 \le j \le k$) represents the number of flows with different sizes, the maximum number of forwarding path built is restricted to 5, k is the maximum number of flow kinds, $B_{i,j}$ represents the bandwidth of different flows; 2) the estimated path lifetime of each path L_i, which is the lifetime of the link with the smallest lifetime of the end-to-end path; 3) the delay of the RREP message D_i between the source and destination nodes. We define the weight W of each path to find a best one among the candidates.

$$W_i = \frac{L_i}{\max(N_{i,j}B_{i,j})D_i} \tag{13}$$

The two best paths are selected from the paths established, according to the routing metric in (13). After building the forwarding paths, the next step is to decide the proportion of traffic each path sustains. In order to avoid the partial congestion, the traffic should be distributed to the network evenly, the two paths should have the same amount of traffic after load balancing. Thus the traffic distributed to the two paths has some relationship with the current traffic along the intermediate nodes of each path.

The maximum amount of traffic along each path, $T_i = \max(N_{i,j}B_{i,j})$, is used there. Assume the route discovery process has built two path, path 1 and path 2, T_1 and T_2 can be got, thus the traffic distributed to path 1 and path 2 will be $T_2/(T_1+T_2)$ and $T_1/(T_1+T_2)$.

V. Numerical Results

The OPNET simulation platform is used to evaluate the performance of our proposed mechanisms, where different flow sizes, two kinds of flows are used, which are 8 Kbps and 32 Kbps to represent the voice and the video services. The duration of each flow is exponentially distributed with mean 100s, the arrival of the flows is Poisson distribution with mean 5.

The sparse scenario is employed to evaluate the performance of our mechanism under different node velocities. The scenario contains a 1000 m×1000 m square field, and the number of nodes is 40. The nodes are distributed randomly and the transmission range for each node is 250 m. We choose the Random Walk model in our simulations, and the maximum node speed is 5 m/s. The simulation time is 500 s.

The performance of AODV, TSDRP and the Load Balancing mechanism with link Lifetime Estimation (LBLE) are evaluated under different ratio of movement parameters changing.

A. Bloom Filter Parameters

The parameters of Bloom filter are directly related to each other. Actually, the optimal combination of m and k depends on the network diameter of the application scenario: for given m, the optimal k for a minimum false positive rate is not the same for each scenario. The analytical results regarding the false positives provide some hints. For given Bloom filter length m and the number of nodes n within the scenario, the false positive rate is minimized for

$$k = \frac{m \ln 2}{n} \tag{14}$$

The local standard filter needs not to be shared among nodes, so the length of the bloom filter can be large. We choose m=256 and n_{flow}=30 since the number of flows of each node will not be too large, then we have k=5.91. Since k is an integer, we choose k=6. The false positive rate is calculated to be 1.66% according to (2), which is neglectable. If we choose n_{flow}=20, according to (6), k=9, and the false positive rate is 0.213%, which is much less than the previous case, but the number of hash functions is 9, the calculation burden for each node will be aggravated, thus we choose the n_{flow}=30 and k=6 case.

B. The Accuracy of Link Lifetime Estimation Mechanism

As mentioned above, the interval of link lifetime calculation influences the accuracy of the estimated result. In order to evaluate the effect of calculation intervals, we use several interval values in our simulation. The interval values are 0.25 s, 0.5 s and 1 s, and the node speeds are 0.5 m/s, 1.5 m/s and 2.5 m/s. Fig. 3 illustrates the average error ratios between the estimated link lifetime and the simulations. Here (0.25, 0.5) means that the calculation interval is 0.25 s and the node speed is 0.5 m/s. The multi-path reflection effect and noise affect the received signal strength value and make the estimation result inaccurate. Some noise is added to the simulations to provide a more real scenario.

In Fig. 3, it can be seen that if the calculation interval time is 1 s, the average error ratio under different node speeds is up to 35.5%, that is because the calculation interval is too large, and the nodes change their moving states so frequently that the moving status cannot be reflected in time by using this sample interval. If the sample interval is 0.25 s, the average error ratio is about 28.7%. With a shorter sample interval, the effect of the radio receiver errors is amplified, and small calculation interval causes large computation load. The 0.5 s case performs the best among the three cases, in which the average error ratio is about 22%, since this case balances the estimation accuracy and the computation load. Therefore, the sample interval is chosen to be 0.5 s in the following simulations. Due to random node mobility, the estimated link lifetime value may have errors; if we increase the number of simulations, such as 10000 times, the error will be reduced greatly. The fluctuations between the estimated results and the simulations somewhat reflect the nature of link instability and it is difficult to match them theoretically. Thus the estimated link lifetime is only used as one of the routing metric.

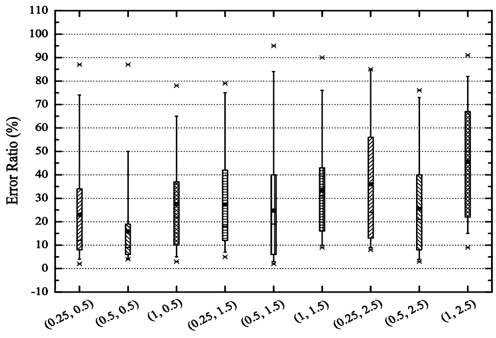

Fig. 3. Error ratios of link lifetime estimation mechanism (%)

C. Forwarding Path Establishing Proportion

This metric is used to reflect the forwarding path establishing ability of different mechanisms. According to the description above, our load balancing mechanism employs the estimated link lifetime as part of the metric to select the paths, thus LBLE can use the paths with longer lifetime than that of TSDRP (in Fig. 4), the network performance can be better 21% than TSDRP mechanisms.

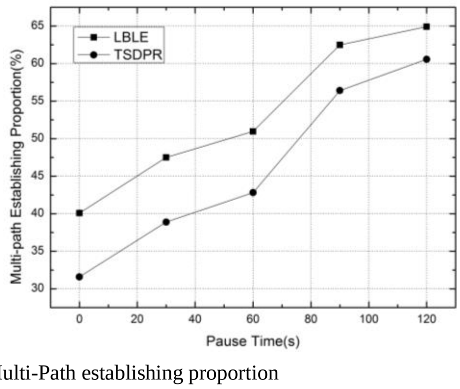

Fig. 4. Multi-Path establishing proportion

D. Packet Delivery Ratio

Fig. 5 illustrates the packet delivery ratio under different pause time. It is obvious that, when using LBLE, the source node can find the paths with less traffic than the shortest path, the partial congestion can be mitigated and the collision times of network will be reduced; moreover, the traffic can be distributed evenly into the network. Since LBLE has more probability of establishing forwarding path than TSDRP, the end-to-end path reliability is higher. Considering that our load balancing mechanisms take the estimated link lifetime as part of the routing metric, the path reliability is higher than that of the other two mechanisms, thus the delivery ratio of our load balancing mechanisms are higher 10% and 5% than that of AODV and TSDRP.

Since the node density is sparse, each node has few neighbours, node mobility can cause frequent link breakages. Although link lifetime is considered during route discovery, link breakages cannot be avoided. As the pause time increases, the link breakage probability decreases, thus the packet delivery ratios of the three mechanisms all increase.

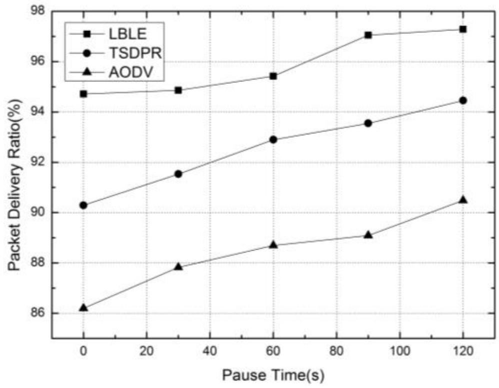

Fig. 5. Packet delivery ratio

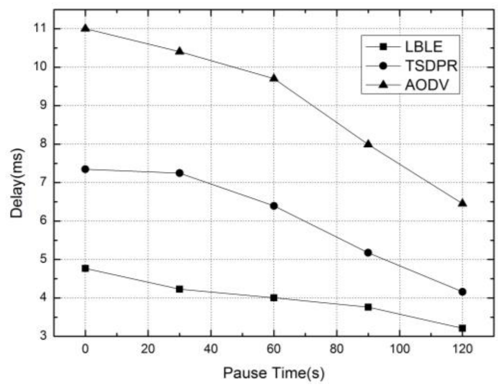

Fig. 6. End-to-End delay

E. End-to-End Delay

In Fig. 6, the end-to-end delay is improved by our load balancing mechanism under the sparse scenario. Since the traffic can be distributed evenly in the network and the forwarding paths are with higher lifetimes than the shortest path in AODV and the paths in TSDRP, the probability of path breakages is less than AODV and TSDRP, data packets need not to wait for re-routing frequently; at the meanwhile, the path broken ratio of LBLE is less than that of the other two mechanisms, less packets need to be retransmitted in the MAC layer, thus the end-to-end delay of our load balancing mechanisms are less 54% and 31% than AODV and TSDRP.

F. Route Discovery Frequency

In Fig. 7, the numbers of RREQs in each scenario are shown. LBLE employs multiple paths to enhance the end-to-end path reliability, the path broken probability is less 85% and 42% than that of AODV and TSDRP, and thus LBLE needs less number of route discovery processes. With the pause time increasing, the path broken probability decreases, less amount of re-routing is needed.

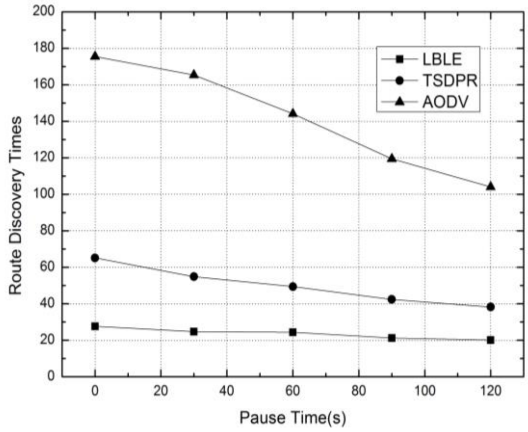

Fig. 7. Route discovery frequency

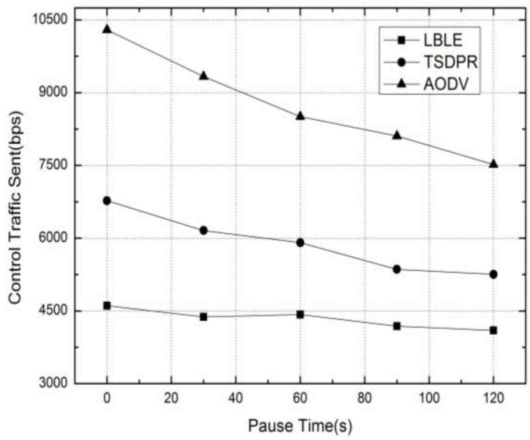

Fig. 8. Control traffic sent

G. Control Traffic Sent

Fig. 8 shows the relationship between the amount of control packets and the pause time. The control packets include the RREQ, RREP, Route Error (RERR) and the Hello messages, in order to monitor the network topology in dynamic scenario in time, HELLO messages are sent periodically. The HELLO messages are sent only since there is no active route at the sending node. The amount of the control messages has direct relationship with the amount of RREQ messages. Since our mechanism balances the traffic evenly in the network, and less RREQs will be sent by the load balancing mechanism, the amount of control traffic is less than that of AODV. Since LBLE uses the paths which last longer than that of TSDRP and the time proportion of multi-path routing is greater than that of TSDRP. Control messages to maintain the multiple path of LBLE is less 45% and 25% than that of AODV and TSDRP, thus the amount of control traffic is the smallest.

VI. CONCLUSIONS

In this paper, a novel load balancing mechanism is introduced to MANET. During the route establishing process, the routing metric combines the estimated path lifetime, the end-to-end delay and the current traffic conditions along the forwarding paths. The Bloom filter is used to identify the flows via each intermediate node. Simulations show that by using our load balancing mechanism, the network performance have been enhanced, such as the packet delivery ratio and the end-to-end delay; the amount of the control packets is restricted since the network condition is better. Implementing our mechanism to the real-world testbed is our future work.

REFERENCES

[1] M. Conti and S. Giordano, “Multihop ad hoc networking: The theory,” *IEEE Communications Magazine*, vol. 45, no. 4, pp. 78-86, 2007.

[2] D. P. Wu, Y. Y. Wang, H. G. Wang, B. R. Yang, C. G. Wang, and R. Y. Wang, “Dynamic coding control in social intermittent connectivity wireless networks,” *IEEE Trans. Veh. Technol.*, no. 99, pp. 1-13, 2005.

[3] M. S. Antesar, P. D. Keshav, K. B. Sanat, and U. A. Irfan, “Recommendation based trust model with an effective defence scheme for MANETs,” *IEEE Trans. Mobile Comput.*, vol. 14, no. 10, pp. 2101-2115, 2015.

[4] H. Y. Kim, H. Kim, Y. H. Cho, and S. H. Lee, “Self-Organizing spectrum breathing and user association for load balancing in wireless networks,” *IEEE Transactions on Wireless Communications*, vol. 15, no. 5, pp. 3409–3421, 2016.

[5] D. P. Wu, H. P. Zhang, H. G. Wang, C. G. Wang, and R. Y. Wang, “Quality-of-protection-driven data forwarding for intermittently connected wireless networks,” *IEEE Wireless Communications*, vol. 22, no. 4, pp. 66-73, 2015.

[6] T. Murakami, I. Sasase, and M. Bandai, “Split multi-path routing protocol with load balancing policy (SMR-LB) to improve TCP performance in mobile ad hoc networks,” in *Proc. 16th IEEE International Symposium on PIMRC*, Berlin, 2005, pp. 1424-1428.

[7] B. Karaoglu and W. Heinzelman, “Cooperative load balancing and dynamic channel allocation for cluster-based mobile ad hoc networks,” *IEEE Trans. Mobile Comput.*, vol. 14, no. 5, pp. 951-963, 2015.

[8] V. Loscri and S. Marano, “A new geographic multipath protocol for ad hoc networks to reduce the route coupling phenomenon,” in *Proc. 63th IEEE Vehicular Technology Conference*, Melbourne, Vic., 2006, pp. 1102-1106.

[9] D. P. Wu, J. He, H. G. Wang, C. G. Wang, and R. Y. Wang, “A hierarchical packet forwarding mechanism for

energy harvesting wireless sensor networks," *IEEE Communications Magazine*, vol. 53, no. 8, pp. 92-98, 2015.

[10] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *ACM Commun.*, vol. 13, no. 7, pp. 442-426, 1970.

[11] L. Pelusi, A. Passarella, and M. Conti, "Opportunistic networking: Data forwarding in disconnected mobile ad hoc networks," *IEEE Communications Magazine*, vol. 44, no. 11, pp. 134–141, 2006.

[12] Y. Li, Y. Jiang, D. Jin, L. Su, L. Zeng, and D. O. Wu, "Energyefficient optimal opportunistic forwarding for delay-tolerant networks," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 9, pp. 4500–4512, 2010.

[13] S. J. Lee and M. Gerla, "Dynamic load-aware routing in ad hoc networks," in *Proc. IEEE International Conference on Communications*, Helsinki, 2001, pp. 3206-3210.

[14] Y. J. Lee and G. F. Riley, "A workload-based adaptive load-balancing technique for mobile ad hoc networks," in *Proc. IEEE Wireless Communications and Networking Conference*, 2005, pp. 2002-2007.

[15] S. Y. Yin and X. K. Lin, "Adaptive load balancing in mobile ad hoc networks," in *Proc. IEEE Wireless Communications and Networking Conference*, 2005, pp. 1982-1987.

[16] P. Engelstad, Y. Zheng, T. Jonvik, and D. Van Thanh, "Trust based secure on demand routing protocol (TSDRP) for MANETs," in *Proc. 4th IEEE International Conference on Advanced Computing & Communication Technologies*, Rohtak, 2014, pp. 432-438.

[17] C. Perkins, E Royer, and S. Das, "Ad hoc on-demand distance vector (AODV) routing," RFC 3561, July 2003.

Pei-Yun Luo was born in Chongqing Province, China, in 1978. She has obtained the master degree. She now works as an associate director in Electronic material evidence forensic center, Chongqing Municipal Public Security Bureau, China. Her research interests include network security, information technology and electronic data forensics.