Ad Hoc Networks 63 (2017) 104–114

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

An efficient protocol for load-balanced multipath routing in mobile ad hoc networks

Ansuman Bhattacharya[a], Koushik Sinha[b,*]

[a] Department of Computer Science and Engineering, National Institute of Technology Meghalaya, Shillong - 793003, India
[b] Department of Computer Science, Southern Illinois University, Carbondale, IL 62901, USA

ARTICLE INFO

Article history:
Received 6 July 2016
Revised 18 April 2017
Accepted 29 May 2017
Available online 30 May 2017

Keywords:
Mobile Ad hoc NETwork (MANET)
Point-to-point routing
Ad hoc On-demand Distance Vector (AODV) routing
Fibonacci Multipath Load Balancing (FMLB) protocol
Load-balanced multipath routing

ABSTRACT

In this paper, we propose a new routing protocol called the *Least Common Multiple based Routing* (*LCMR*) for load-balanced multipath routing in *Mobile Ad hoc NETworks* (*MANETs*). First, we find multiple paths between a source to a destination, when those exist, along with the estimates of the time to route a packet along each of these paths. The data packets originating from the source to the destination are then distributed along these multiple paths in such a way that the number of data packets sent along any such path is inversely proportional to the routing time through this path. This distribution strategy keeps the load balanced along all the paths so that the overall routing time for sending the data packets is minimized. Routes between a given source-destination pair are discovered in a way similar to that in the *Ad hoc On-demand Distance Vector* (*AODV*) routing protocol with the difference that instead of the number of hops, the routing time for reaching the destination along every route is measured, and multiple routes, if those exist, will also be determined by the route discovery process. Our proposed technique for distribution of packets along different routes is very elegant with a better performance than the existing load-balanced routing protocols like *Fibonacci Multipath Load Balancing* (*FMLB*) and *Multiple AODV* (*MAODV*), as established from a theoretical analysis as well as through simulation results.

1. Introduction

Routing in *Mobile Ad hoc NETworks* (*MANETs*) is a well studied topic [1–9]. The *Ad hoc On-demand Distance Vector* (*AODV*) [3] routing is one of the most popular and widely used protocols in *MANETs*. Routing with congestion awareness and adaptivity has also been considered by some researchers [10]. Recently, a protocol has been proposed in [11] which is based on estimating the routing time (instead of the number of hops, as in *AODV*) along different paths during the route discovery phase and then choosing the route with the minimum routing time. Such routing protocol takes into consideration the link delay based on the channel capacity as well as the queuing delay due to congestion at the intermediate nodes. Note that routing time may be more along a path, say, P_1 having smaller number of hops than that along a path P_2 with more number of hops if there is more congestion at the nodes on the path P_1 than that at the nodes on the path P_2.

Another class of routing protocol in *MANETs* is based on finding multiple possible paths between a given source-destination pair and then distributing the total set of data of packets from the source to the destination along all such possible paths. Apart from the reduced time of routing all the message packets through multiple paths, another advantage of multipath routing is to increase the reliability of communication. For example, let us consider two node-disjoint paths with reliability r_1 and r_2, respectively, $0 \leq r_1, r_2 \leq 1$, which are defined to be the probabilities that the message is received correctly along the respective paths. Then the combined reliability of communication through both these paths is given by $1-(1-r_1)(1-r_2) = r_1+r_2-r_1r_2$ which will be more than either of r_1 and r_2. For example, if $r_1 = 1-10^{-3} = 0.999$ and $r_2 = 1-10^{-4} = 0.9999$, then the combined reliability will be given by $1-10^{-7} = 0.9999999$.

Fibonacci sequence based Multipath Load Balancing (*FMLB*) is one such multipath routing protocol [12] which distributes the total load (i.e., the packets) in different paths following the numbers in the Fibonacci sequence for balancing the load along different paths. To be specific, consider the situation as shown in Fig. 1 with S and D as the source and the destination nodes. Between S and D, there are three possible paths: *i*) path P_1 through nodes u_1, u_2 and u_3 with T_1 as the routing time, *ii*) path P_2 through nodes u_4, u_5 and u_6 with T_2 as the routing time, and *iii*) path P_3 through nodes u_7 and u_8 with T_3 as the routing time. Assume that due to the

* Corresponding author.
E-mail addresses: ansuman@nitm.ac.in (A. Bhattacharya), koushik.sinha@cs.siu.edu, sinha_kou@yahoo.com, koushik.sinha@gmail.com (K. Sinha).

http://dx.doi.org/10.1016/j.adhoc.2017.05.008
1570-8705/

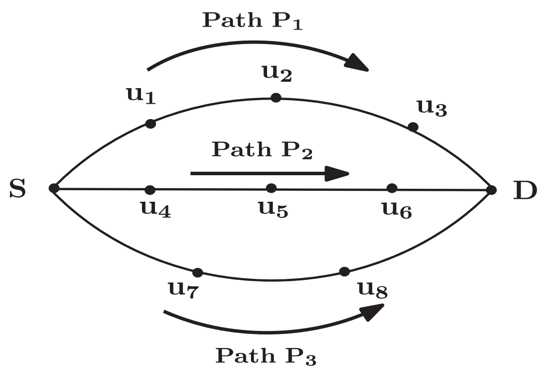


Fig. 1. An example for multiple paths between source *S* and destination *D*.

differences in channel capacity and congestion at various nodes, $T_1 > T_2 > T_3$. Following the *FMLB* technique as presented in [12], we take the three consecutive Fibonacci numbers F_1, F_2 and F_3 so that for every $F_1 + F_2 + F_3 = 4$ packets, $F_1 = 1$ packet along the path with the maximum routing time T_1, $F_2 = 1$ packet along the path with the next maximum routing time T_2 and $F_3 = 2$ packets along the path with the minimum routing time T_3 are routed. The overall time for routing all these four packets is *max* $(T_1, T_2, 2T_3)$.

1.1. Related works

Multipath routing in MANETs has been extensively studied in the last two decades. Apart from the works already mentioned above [10–12], several other results exist in the literature for routing via multiple paths in ad hoc networks. We present below some of those important works.

Ganjali and Keshavanian [13] introduced a model for evaluating load balancing in multi-path routing, where the first *K* shortest paths (for a pre-specified *K*) are chosen and then the load is distributed uniformly among those paths. Pearlman et al. [14] demonstrated the impact of route coupling on *Alternate path routing*'s (*APR*'s) delay performance in ad-hoc networks. In multiple channel environments, *APR* is able to provide a 20% reduction in end-to-end delay for bursty data streams. Wang et al. [15] proposed a multipath routing protocol for ad hoc wireless networks *Multi-path Source Routing* (*MSR*), which is an extension of *Dynamic Source Routing* (*DSR*). Based on the measurement of *RTT* tool, they proposed a scheme to distribute load among multiple paths. They demonstrated that *MSR* decreases the network congestion and increases the path fault tolerance quite well. Parissidis et al. [16] presented the results of a detailed simulation study of three multipath routing protocols *Split Multi-path Routing* (*SMR*), *Ad hoc On demand Multi-path Distance Vector routing* (*AOMDV*) and *Ad hoc On-demand Distance Vector Multi-path* (*AODV_Multipath*). Their simulation study shows that the *AOMDV* protocol achieves the best performance in high mobility scenarios, while *AODV_Multipath* performs better in scenarios with low mobility and higher node density. *SMR* performs best in networks with low node density. However, as node density increases, the protocols performance is degraded. Pham and Perreau [17] analyzed and compared reactive single path and multipath routing with load balancing mechanisms in ad hoc networks, in terms of overhead, traffic distribution and connection throughput. Their results reveal that compared to general single path routing protocol, multipath routing mechanism requires more overhead but leads to better performance in terms of congestion and capacity, provided the route length is within a certain upper bound. The analytical results are further confirmed by simulation. Nasipuri et al. [18] focused on *Dynamic Source Routing* (*DSR*) and they showed how an intelligent use of multipath routing technique can reduce the frequency of query floods. They developed an analytic modeling framework to determine the relative frequencies of query floods under various techniques. Their modeling showed that while multipath routing is significantly better than single path routing, the performance advantage is small beyond a few paths and for long path lengths. They also showed that providing all intermediate nodes in the primary (shortest) route with alternative paths had a significantly better performance than providing only the source with alternate paths.

1.2. Our contribution

We propose here a load-balanced multipath routing protocol called *Least Common Multiple based Routing* (*LCMR*) in mobile ad hoc networks with balanced traffic load along different possible paths. Multiple paths between a given source-destination pair, when they exist, are first discovered with estimate of routing time through each of the paths. The data packets from the source to the destination are then distributed along these multiple paths in such a way that the number of data packets sent along any such path is inversely proportional to the routing time through this path. This distribution strategy keeps the load balanced along all the paths so that the overall routing time for sending the data packets is minimized. Following this strategy, to find the number of data packets to be sent along a path, we compute the least common multiple (*L*) of the routing time along all the paths between the source-destination pair. Dividing this *L* by the routing time along a path, we get the relative number of packets to be sent along that path. This will make the load balanced as all the data packets can be sent through the multiple paths with almost equal routing time along each path. As an example, let there be three paths P_1, P_2 and P_3 from a source to a destination with routing time as 40, 30 and 20 units of time, respectively. We find the least common multiple of 40, 30 and 20 as 120. We divide 120 by 40, 30 and 20 to get 3, 4 and 6, respectively. Hence, for every $3 + 4 + 6 = 13$ data packets, we send 3 data packets along the path P_1, 4 data packets along P_2 and 6 data packets along P_3 so that the routing time along each of the paths is 120 time units. We note that this problem definition is similar to the problems of scheduling *N* independent tasks on *k* uniform or unrelated parallel processors [19–21], which are known to be $NP-hard$. In this paper, we show both theoretically and through simulation results that the proposed *LCMR* multipath routing protocol performs better than the multipath *FMLB* protocol [12]. The intuitive reason behind this improvement in routing time by our proposed *LCMR* protocol is that our load distribution technique takes care of the actual routing time along different paths, while that in the *FMLB* protocol has not taken this into consideration. The ratios of actual routing time along different possible paths may be much different from the ratios of the corresponding Fibonacci numbers which do not have any natural correspondence with the routing time along different paths. The proposed *LCMR* routing protocol also performs better than the *Multipath AODV* (*MAODV*) protocol [13] with equal number of packets along each of the paths, as in *LCMR* we send more number of packets (assuming all packets to be of the same size) along paths requiring less routing time instead of routing equal number of packets along all the paths.

2. Proposed multipath routing protocol

The *LCMR* load-balanced multipath routing protocol is based on first finding several possible routes from the source to the destination and the associated time of flight (i.e., routing time) in each path. Routes between a given source-destination pair are discovered in a way similar to that in the *AODV* routing protocol with the difference that instead of the number of hops, the routing time for reaching the destination along every route is measured, and multiple routes, if those exist, will also be determined by the route discovery process. Although the network topology as well as the traffic in the network may change in a dynamic situation, we assume that this route discovery process is completed fast enough so that during this route discovery, neither the network topology nor the traffic in the network changes. First, the source node issues a route request message *RREQ* with source *id*, destination *id*, *RREQ* sequence number, time of generation of the *RREQ* message and the list of intermediate nodes traversed. The destination node, on receiving the *RREQ* message, sends the route reply message *RREP* back to the source node. After receiving the *RREP* message, the source node not only identifies the route, but also the time taken by a packet for the forward trip along this route from source to destination. To put a limit on such possible paths, the source node may wait for a maximum time out period, say, T_{max}, to get all the *RREP* messages coming along different routes corresponding to the given sequence number of the *RREQ* message.

Suppose, k number of possible paths $P_1, P_2, \ldots, P_k$ are discovered with estimated routing time as $T_1, T_2, \ldots, T_k$, respectively such that $T_1 \geq T_2 \geq \cdots \geq T_k$. Let L = *Least Common Multiple* (*LCM*) of T_1, $T_2, \ldots$ and T_k.

Let $n_i = \frac{L}{T_i}$, $\forall i$, $1 \leq i \leq k$, and $n = \sum_{i=1}^{k} n_i$. Out of n consecutive data packets, we propose to schedule n_i data packets along the path P_i. That is, the data packets are sent along the paths $P_1, P_2, \ldots, P_k$ in the ratio of $n_1 : n_2 : \cdots : n_k$. Thus, the total time of routing n data packets along these k paths is given by $max(n_i T_i) = L$.

If the total number of data packets to be sent is $N = pn + r, 0 \leq r < n$ for some integer $p \geq 0$, then the first pn packets are sent along the paths $P_1, P_2, \ldots, P_k$ in the ratio of $n_1 : n_2 : \cdots : n_k$ and the remaining r packets are sent as follows: $\lfloor \frac{r}{n} \times n_1 \rfloor$ packets along P_1, $\lfloor \frac{r}{n} \times n_2 \rfloor$ packets along P_2, $\cdots$, $\lfloor \frac{r}{n} \times n_k \rfloor$ packets along P_k, and then all the remaining $(r - \sum_{i=1}^{k} \frac{r}{n} \times n_i)$ packets along the path P_k only.

To implement the above protocol, we now describe below the algorithms to be executed by the source node (Algorithm 1), the intermediate nodes (Algorithm 2) and destination node (Algorithm 3).

Algorithm 1: Find_Route_Source_Node.

Input: *DA*
Output: *RREQ*, Data packets
if *SA = its own id* **then**
  `/* Path initialization */`
  Initialize a real time clock T to 0;
  Broadcast *RREQ* message with its *SA* and *DA* ;
  while $(T < T_{max})$ **do** `// ` T_{max} ` is the time-out period`
    if *RREP message received* **then**
      Collect the *RREP* messages and create a path list with P_i and T_i ;
    $T = T + 1$;
  if *Path list is created* **then**
    Calculate L from all T_i ;
    Calculate $n_i = \frac{L}{T_i}$;
  `/* Data Transmission */`
  while *all data packets are not sent* **do**
    Send Data packet and wait for δ_T time ; `// according to ratio of ` n_i ` values`
    if *ACK received within* δ_T *time* **then**
      Send next Data packet ;
    else
      Resend Data packet ;

Algorithm 2: Find_Route_Intermediate_Node.

Input: Routing Messages Received
Output: Routing Messages Transmitted
if *both SA and DA not equal to its own id* **then**
  if *RREQ received* **then**
    Send *RREQ* message with its own *id* ;
  else if *RREP received* **then**
    Send *RREP* message with its own *id* ;
  else if *Data packet received* **then**
    Forward data packet ;
  else if *ACK packet received* **then**
    Forward *ACK* packet ;
  else if *NACK packet received* **then**
    Forward *NACK* packet ;

Algorithm 3: Find_Route_Destination_Node.

Input: *RREQ*, Data packet
Output: *RREP*, *ACK*, *NACK*
if *DA = its own id* **then**
  `/* Path initialization */`
  if *RREQ received* **then**
    Send *RREP* message from which *RREQ* received ;
  `/* Data Transmission */`
  if *Data packet received properly* **then**
    Send *ACK* packet ;
  else
    Send *NACK* packet ;

It may be noted that Algorithm 1 computes the multipaths for routing the packets which may not be, in general, edge-disjoint or node-disjoint.

3. Performance evaluation

In this section, we theoretically analyze the performance of the *LCMR* protocol based on the assumption that all the multipaths used for routing the packets are node-disjoint. As already mentioned above, in practice, these paths may not be always node-disjoint or even edge-disjoint, in which cases the routing time will be more than the theoretically derived value. We consider, however, those situations through simulation, as presented in a later section.

For the theoretical analysis, we first consider the simplest case of routing through only two possible paths. Without loss of generality, let us assume that N, the number of data packets to be sent from S to D is even. We now have the following result.

Theorem 1. *When there are only two possible paths P_1 and P_2 for routing with unequal routing time T_1 and T_2, $T_1 > T_2$, the routing time R_p with the proposed LCMR protocol will always be less than the routing time R_f with the FMLB protocol.*

Proof. Let $N = pn + r, 0 \leq r \leq n - 1$. Then using *LCMR* protocol, $\lfloor n_1 N/n \rfloor = pn_1 + \lfloor \frac{r}{n} \times n_1 \rfloor$ packets will be routed through P_1, while $pn_2 + \lceil \frac{r}{n} \times n_2 \rceil$ will be the number of packets routed through P_2.

Hence, $R_p = \max[pn_1 T_1 + \lfloor \frac{r}{n} \times n_1 \rfloor T_1, \;\; pn_2 T_2 + \lceil \frac{r}{n} \times n_2 \rceil T_2]$ and $R_f = pT_1(n_1 + n_2)/2 + \frac{r}{2} \times T_1$, as $n_1 + n_2 = N$ is assumed to be even and $T_1 > T_2$.

Since $n_1 < n_2$, $\frac{n_1}{n} < \frac{1}{2} < \frac{n_2}{2}$. Hence, $\lceil \frac{r}{n} \times n_2 \rceil > \frac{r}{2}$ and $\lfloor \frac{r}{n} \times n_1 \rfloor < \frac{r}{2}$. Since, $\frac{T_2}{T_1} = \frac{n_1}{n_2}$, we can say, $\frac{T_2}{T_1 + T_2} = \frac{n_1}{n_1 + n_2}$. Thus, $T_2 = \frac{n_1}{n} \times$

$(T_1 + T_2)$. Now,

$$\left\lceil \frac{r}{n} \times n_2 \right\rceil T_2 = \left\lceil \frac{r}{n} \times n_2 \right\rceil \frac{n_1}{n}(T_1 + T_2) \geq \frac{r}{n}.\frac{n_1 n_2}{n}.(T_1 + T_2) \quad (1)$$

Similarly,

$$\left\lfloor \frac{r}{n} \times n_1 \right\rfloor T_1 = \left\lfloor \frac{r}{n} \times n_1 \right\rfloor \frac{n_2}{n}(T_1 + T_2) \leq \frac{r}{n}.\frac{n_1 n_2}{n}.(T_1 + T_2) \quad (2)$$

Thus, $\lceil \frac{r}{n} \times n_2 \rceil T_2 \geq \lfloor \frac{r}{n} \times n_1 \rfloor T_1$. Also, since $n_1 T_1 = n_2 T_2$, it follows that $R_p = p n_2 T_2 + \lceil \frac{r}{n} \times n_2 \rceil T_2$.

Hence,

$$\begin{aligned} R_f - R_p &= pT_1(n_1 + n_2)/2 + \frac{r}{2} \times T_1 - pn_2 T_2 - \left\lceil \frac{r}{n} \times n_2 \right\rceil T_2 \\ &> p(n_2 T_2 + n_2 T_1)/2 + \frac{r}{2} \times T_1 - pn_2 T_2 - \frac{r}{n} \times n_2 T_2 \\ &= pn_2(T_1 - T_2)/2 + \frac{r}{2} \times T_1 - \frac{r}{n} \times n_1 T_1 \\ &= pn_2(T_1 - T_2)/2 + rT_1\left(\frac{1}{2} - \frac{n_1}{n}\right). \end{aligned}$$

Since $\frac{n_1}{n} < \frac{1}{2}$, $rT_1(\frac{1}{2} - \frac{n_1}{n})$ is greater than zero. Hence, R_f will always be greater than R_p.

Hence the claim. □

Remark 1. When there are only two possible paths for routing from S to D, the *FMLB* protocol and the *MAODV* protocol [13] (with equal distribution of the number of data packets along the two paths) yield the same result. Hence, with only two possible paths having unequal routing time, the *LCMR* protocol always requires smaller routing time than either of the *FMLB* protocol and the *MAODV* protocol.

To evaluate the performance of our proposed *LCMR* protocol in the general case with two or more paths from S to D, let us first consider a few examples as given below.

Example 1. Consider the scenario as given in Fig. 1 and assume that the values of T_1, T_2, T_3 along the three paths P_1, P_2 and P_3 are given by 40, 30 and 20 units of time, respectively. The least common multiple L of 40, 30 and 20 is 120. Since $\frac{120}{40} = 3$, $\frac{120}{30} = 4$ and $\frac{120}{20} = 6$, data packets are sent along P_1, P_2 and P_3 in the ratio of 3: 4: 6. Thus, if we have $6 + 4 + 3 = 13$ data packets, we send three packets along P_1, four packets along P_2 and six packets along P_3 so that the overall routing time for sending all these thirteen packets is only 120 time units.

For the same scenario, if we have used *FMLB* protocol, then the number of data packets along the paths P_1, P_2 and P_3 would be in the ratio of F_1: F_2: F_3, i.e., 1: 1: 2. Hence, the first twelve packets will be sent with three packets along path P_1, three packets along P_2 and six packets along P_3. The last packet will be sent along P_3 requiring the least routing time of 20 units. Thus, the total routing time for the packets along P_1 is 120 units, along P_2 is 90 units and along P_3 is 140 units. That is, the total routing time is 140 units which is more than that in our proposed *LCMR* protocol.

If one would have used the *MAODV* protocol [13] with equal distribution of packets along each of the three paths (with one extra packet after equal distribution being routed through P_3 which has the minimum routing time), then the corresponding routing time would have been $max(160, 120, 100) = 160$ units, which is more than that in our proposed *LCMR* protocol.

Example 2. Assume that using the same paths as in Fig. 1, we need to send a total of 100 data packets from S to D. We note that $n = 13$ and $100 = 7 \times 13 + 9$. Hence, following our proposed *LCMR* protocol, we first send 91 data packets with $7 \times 3 = 21$ packets along P_1, $7 \times 4 = 28$ packets along P_2 and $7 \times 6 = 42$ packets along P_3.

Noting that $\lfloor \frac{9}{13} \times 3 \rfloor = 2$, $\lfloor \frac{9}{13} \times 4 \rfloor = 2$, and $\lfloor \frac{9}{13} \times 6 \rfloor = 4$, out of the remaining 9 packets, 2 packets are sent along P_1, 2 packets along P_2 and 4 packets along P_3. Finally the last packet is sent along P_3.

Thus, the number of packets sent along P_1 is $21 + 2 = 23$ with a routing time of 920 units, that sent along P_2 is $28 + 2 = 30$ with a routing time of 900 units and that sent along P_3 is $42 + 5 = 47$ with a routing time of 940 units. Hence, the resulting routing time for sending all the 100 data packets along the three paths is $max(920, 900, 940) = 940$ time units.

However, following the *FMLB* protocol [12], the packets will have been distributed in the ratio of 1: 1: 2, i.e., 25 packets along each of P_1 and P_2, and 50 packets along P_3, leading to a total routing time of $50.20 = 1000$ units, which is more than that in our proposed LCMR protocol.

If one would have used the *MAODV* protocol [13] with equal distribution of packets along each of the three paths, then the corresponding routing time would have been $max(1320, 990, 680) =$ 1320 units, which is more than that in our proposed *LCMR* protocol.

Remark 2. Note that in Example 2, the minimum time required to route all the 100 data packets from S to D using the three paths P_1, P_2 and P_3 would be 840 time units for the first 91 packets and 90 time units for the remaining 9 packets (2 packets sent along P_1, 3 packets along P_2, and 4 packets along P_3) for a total of 930 time units. As stated earlier, finding the minimum routing time is similar to the NP-hard problem of scheduling N independent tasks on k parallel processors [19,20].

Now consider the general case that there are k paths P_1, P_2,..., P_k with routing time T_1, T_2,..., T_k such that $T_1 \geq T_2 \geq \cdots \geq T_k$. For the sake of simplicity, we assume in our later discussions that the total number N of data packets to be routed from S to D is an integral multiple of both $\sum_{i=1}^{k} n_i$ and $\sum_{i=1}^{k} F_i$.

The routing time R_p for routing all the N data packets from S to D through the k paths P_1, P_2,..., P_k. following our proposed *LCMR* protocol is given by $R_p = max_i[\frac{Nn_i}{\sum_{i=1}^{k} n_i} \times \frac{L}{n_i}] = \frac{LN}{\sum_{i=1}^{k} n_i}$.

On the other hand, the routing time R_f following *FLMB* is given by $R_f = max_i[\frac{NF_i}{\sum_{i=1}^{k} F_i} \times \frac{L}{n_i}] = max_i[\frac{LNF_i}{n_i \sum_{i=1}^{k} F_i}]$.

We now have the following result.

Theorem 2. *The time for routing N data packets with our proposed LCMR protocol will always be smaller than that in FMLB protocol [12] whenever the ratio n_1: n_2: $\cdots$: n_k will be different from F_1: F_2: $\cdots$: F_k.*

Proof. Let $\frac{F_i}{n_i} = r_i$ and suppose, $r_{i_1} \geq r_{i_2} \geq \cdots r_{i_k}$, where $(i_1, i_2,..., i_k)$ is some permutation of $(1, 2,..., k)$. Hence, $\frac{F_1+F_2+\cdots+F_k}{n_1+n_2+\cdots+n_k} = \frac{n_1 r_{i_1} + n_2 r_{i_2} + \cdots + n_k r_{i_k}}{n_1+n_2+\cdots+n_k}$. Since, n_1: n_2: $\cdots$: n_k is not equal to F_1: F_2: $\cdots$: F_k (by proposition), all of r_{i_1}, r_{i_2}, $\cdots$, r_{i_k} cannot be equal. Hence,

$$\begin{aligned} \frac{F_1 + F_2 + \cdots + F_k}{n_1 + n_2 + \cdots + n_k} &< \frac{n_1 r_{i_1} + n_2 r_{i_1} + \cdots + n_k r_{i_1}}{n_1 + n_2 + \cdots + n_k} \\ &< \frac{r_{i_1}(n_1 + n_2 + \cdots + n_k)}{n_1 + n_2 + \cdots + n_k} = r_{i_1} = max_i\left[\frac{F_i}{n_i}\right] \end{aligned}$$

Now, from the above expressions of R_f and R_p, it follows that $\frac{R_f}{R_p} = max_i[\frac{F_i}{n_i}] \times \frac{\sum_{i=1}^{k} n_i}{\sum_{i=1}^{k} F_i} > 1$. Hence the proof. □

We illustrate the result with the following examples.

Example 3. Let $N = 300$ packets are to be sent from S to D along six possible paths P_1, P_2, P_3, P_4, P_5 and P_6 arranged in the order of non-increasing routing time. The corresponding distribution of packets following the *FMLB* protocol will be in the ratio of the Fibonacci numbers F_1, F_2, F_3, F_4, F_5 and F_6, i.e., 1: 1: 2: 3: 5: 8 along

Table 1
Summary of parameters.

Number of nodes	25	
Size of network	$(50 \times 50)\,m^2$ for random network $(40 \times 40)\,m^2$ for regular grid network	
Communication range of a node	15 m for random network 10 m for regular grid network	
Size of a packet	1KB	
Bandwidth of each channel	1 MBps	
Link delay	1 msec	
Random traffic generation	Time of generation Source and destination nodes Mean interarrival time	Poisson distribution, Uniform random numbers from 1 to 25, Varied from 2 s to 20 s
Number of packets generated at each request	Varied from 2,000 to 10,000	

P_1, P_2, P_3, P_4, P_5 and P_6, respectively irrespective of the actual routing time along these paths. That is, the number of packets sent along these paths will be 15, 15, 30, 45, 75 and 120, respectively.

Now assume that the routing time along P_1, P_2, P_3, P_4, P_5 and P_6 paths are such that according to our proposed *LCMR* protocol, the packets along these paths would be distributed in the ratio of 1: 1: 2: 3: 4: 4, respectively. Then the number of packets sent along these paths will be 20, 20, 40, 60, 80 and 80, respectively. Note that in this case, $\sum_{i=1}^{k} F_i = 20$ which is greater than $\sum_{i=1}^{k} n_i = 15$.

Assume that L is the least common multiple of the values of routing time along these six paths. Thus, 15 packets can be sent following our proposed *LCMR* protocol in L time units, i.e., $R_p = 20L$ time units. Routing time R_f is computed as $max(\frac{300\times1\times L}{20\times1}, \frac{300\times1\times L}{20\times1}, \frac{300\times2\times L}{20\times2}, \frac{300\times3\times L}{20\times3}, \frac{300\times5\times L}{20\times4}, \frac{300\times8\times L}{20\times4}) = max(15L, 15L, 15L, 15L, \frac{75}{4}L, 30L) = 30L$. Thus, $R_p < R_f$ and we also note that $max(\frac{F_1}{n_1}, \frac{F_2}{n_2}, \frac{F_3}{n_3}, \frac{F_4}{n_4}, \frac{F_5}{n_5}, \frac{F_6}{n_6}) = 2$ which is greater than $\frac{\sum_{i=1}^{k} F_i}{\sum_{i=1}^{k} n_i} = \frac{20}{15} = 1.33$.

Example 4. In Example 3, consider different routing time along the six paths so that the number of packets following our proposed *LCMR* protocol along these paths are in the ratio of 1: 2: 3: 4: 4: 6, respectively. Hence, $\sum_{i=1}^{k} n_i = 20$ which is equal to $\sum_{i=1}^{k} F_i$. In this case, 20 packets can be sent following our proposed *LCMR* protocol in L time units, i.e., $R_p = 15L$ time units, and R_f is computed as $20L$. Thus, $R_p < R_f$. We also note that $max(\frac{F_1}{n_1}, \frac{F_2}{n_2}, \frac{F_3}{n_3}, \frac{F_4}{n_4}, \frac{F_5}{n_5}, \frac{F_6}{n_6}) = \frac{8}{6} = 1.33$ which is greater than $\frac{\sum_{i=1}^{k} F_i}{\sum_{i=1}^{k} n_i} = \frac{20}{20} = 1$.

4. Simulation results

In this section, we show the results of simulating our proposed *LCMR* protocol in a real-life situation assuming that the multipaths used for routing may not be node-disjoint or edge-disjoint. We compare the performance of the *LCMR* protocol with that of the four existing protocols *FMLBRT* [11], *FLMBHC* [12], *MAODVRT* and *MAODVHC* [13], where *FMLBRT* [11] finds k-shortest paths according to routing time and then distributes the packets along different paths following the numbers in the Fibonacci sequence (with the largest number of packets along the path having the shortest routing time), *FLMBHC* [12] determines k-shortest paths according to hop count and then distributes the packets along different paths following the numbers in the Fibonacci sequence, *MAODVRT* and *MAODVHC* [13] involve finding k-shortest paths based on routing time and hop counts, respectively and then distributing the packets evenly along these paths.

Simulations are performed on random network topologies and also on regular grid networks with 25 nodes each, using *Network Simulator 3* (*NS*3) (version *NS* – 3.26). In random network topology, nodes are distributed randomly following uniform distribution within an area of $(50 \times 50)\,m^2$, while in the regular grid network topology, the nodes are placed on grid points with 10 m as the separation between the grid points along each orthogonal direction within an area of $(40 \times 40)\,m^2$. For random network topology, the communication range of each node has been assumed to be 15 m, and for the regular grid topology, the communication range of each node has been taken as 10 m (Table 1).

We assume for simulation that the traffic in the network is generated in a random manner. To be specific, we refer to the ith traffic generated for sending data packets from a source S_i to a destination D_i by the triplet (t_i, S_i, D_i), where t_i is the time of generation which follows Poisson distribution, S_i and D_i are generated independently by two uniform random number generators in the range 1–25. The number of data packets to be sent from a source to a destination is assumed to be same for all the requests and is varied from 2,000 packets to 10,000 packets, each of 1 KB size (corresponding to a volume of data ranging from around 2 MB–10 MB). The mean interarrival time of these requests following Poisson distribution is varied from 2 s to 20 s. Bandwidth of each channel is assumed to be 1 MBps so that the link delay, i.e., time taken by a packet through one link is 1 msec. Since we assume for this simulation that the multiple paths chosen for a given source-destination pair may be, in general, neither edge-disjoint nor node-disjoint, queues may be formed at a node due to packets arriving at a node from different source-destination pairs as well as for the same source-destination pair (when two paths for a given source destination pair share a common intermediate node). The NS3 simulation environment takes care of this dynamic situation so that the total routing time of a packet through a path will be computed as the sum of all the link delays and the associated queue delays at all the intermediate nodes in that path.

To get the simulation results, for every source-destination pair having j paths in the network, where j varies from 1 to some k, we assume that the packets are transmitted from the source to the destination along these j paths. For the simulation, we have set the value of k as 5.

As a test case of the dynamic behavior of the simulation, we first made an experiment for routing of packets between a given source-destination pair only, with all other nodes being just packet forwarding nodes, on a random network topology with the number of paths varying from 1 to 5. Table 2 shows the results from this simulation experiment where all such paths between the chosen source-destination pair were node-disjoint, while Table 3 shows the results of such experiment with a given source-destination pair

Table 2
Comparison of routing time for a source-destination pair having node-disjoint multipaths.

		LCMR			FMLBRT			MAODVRT				FMLBHC			MAODVHC		
Number of paths	Initial estimated routing time along a path (in msec.)	No. of packets sent	Theore-tically estimated routing time (in sec.)	Simulated routing time (in sec.)	No. of packets sent	Theore-tically estimated routing time (in sec.)	Simulated routing time (in sec.)	No. of Packets sent	Theore-tically estimated routing time (in sec.)	Simulated routing time (in sec.)	Initial estimated routing time along a path (in msec.)	No. of Packets sent	Theore-tically estimated routing time (in sec.)	Simulated routing time (in sec.)	No. of Packets sent	Theore-tically estimated routing time (in sec.)	Simulated routing time (in sec.)
1	5	2634	13,170	13.17	4167	20,835	20.835	2000	10,000	16	6	4167	25,002	25.002	2000	12,000	18
2	6	2195	13,170		2501	15,006		2000	12,000		5	2501	12,505		2000	10,000	
3	7	1881	13,167		1666	11,662		2000	14,000		7	1666	11,662		2000	14,000	
4	8	1645	13,160		833	6664		2000	16,000		8	833	6664		2000	16,000	
5	8	1645	13,160		833	6664		2000	16,000		9	833	7497		2000	18,000	
1	5	3152	15,760	15.764	4286	21,430	21.43	2500	12,500	20	6	4286	25,716	25.761	2500	15,000	20
2	6	2627	15,762		2858	17,148		2500	15,000		5	2858	14,290		2500	12,500	
3	7	2252	15,764		1428	9996		2500	17,500		7	1428	9996		2500	17,500	
4	8	1969	15,752		1428	11,424		2500	20,000		8	1428	11,424		2500	20,000	
1	5	3926	19,630	19.63	5000	25,000	25	3334	16,670	23.331	6	5000	30,000	30	3334	20,004	23.3331
2	6	3271	19,626		2500	15,000		3333	19,998		5	2500	12,500		3333	16,665	
3	7	2803	19,621		2500	17,500		3333	23,331		7	2500	17,500		3333	23,331	
1	5	5455	27,275	27.275	5000	25,000	30	5000	25,000	30	6	5000	30,000	30	5000	30,000	30
2	6	4545	27,270		5000	30,000		5000	30,000		5	5000	25,000		5000	25,000	
1	5	10,000	50,000	50	10,000	50,000	50	10,000	50,000	50	6	10,000	60,000	60	10,000	60,000	60

Table 3
Comparison of routing time for a source-destination pair with multipaths having some common edges.

		LCMR			FMLBRT			MAODVRT				FMLBHC			MAODVHC		
Number of paths	Initial estimated routing time along a path (in msec.)	No. of packets sent	Theore-tically estimated routing time (in sec.)	Simulated routing time (in sec.)	No. of packets sent	Theore-tically estimated routing time (in sec.)	Simulated routing time (in sec.)	No. of packets sent	Theore-tically estimated routing time (in sec.)	Simulated routing time (in sec.)	Initial estimated routing time along a path (in msec.)	No. of packets sent	Theore-tically estimated routing time (in sec.)	Simulated routing time (in sec.)	No. of packets sent	Theore-tically estimated routing time (in sec.)	Simulated routing time (in sec.)
1	3	2703	8.112	12.1635	4167	12.501	18.7515	2000	10	12.6	3	4167	12.501	19.37,655	2000	12	13.3
2	4	2028			2501			2000			4	2501			2000		
3	4	2027			1666			2000			4	1666			2000		
4	5	1621			833			2000			6	833			2000		
5	5	1621			833			2000			5	833			2000		
1	3	3226	9.68	14.517	4286	12.858	19.287	2500	12.5	15.25	3	4286	12.858	19.9299	2500	15	16.625
2	4	2420			2858			2500			4	2858			2500		
3	4	2419			1428			2500			4	1428			2500		
4	5	1935			1428			2500			6	1428			2500>		
1	3	4000	12	18	5000	15	22.5	3334	13.332	18.83145	3	5000	15	23.25	3334	13.332	21.83115
2	4	3000			2500			3333			4	2500			3333		
3	4	3000			2500			3333			4	2500			3333		
1	3	5715	17.145	25.7175	5000	20	26	5000	20	26	3	5000	20	28.25	5000	20	28.25
2	4	4285			5000			5000			4	5000			5000		
1	3	10,000	30	45	10,000	30	45	10,000	30	45	3	10,000	30	46.5	10,000	30	46.5

Table 4
Required time in (*sec.*) for different protocols along with number of paths and number of packets.

Number of Packets	Number of paths	Random Network LCMR	FMLBRT	MAODVRT	FMLBHC	MAODVHC	Grid Network LCMR	FMLBRT	MAODVRT	FMLBHC	MAODVHC
10,000	1	45	45	45	46.5	46.5	53	53	53	54	54
	2	24.1245	26	26	28.25	28.25	30.0722	34.75	34.75	35	35
	3	16.9065	22.5	18.8315	23.25	21.8312	21.8943	26.5	26.8306	27	27.6639
	4	13.2392	19.287	15.25	19.9299	16.625	17.4741	22.7158	21.625	23.1444	22.875
	5	10.9408	18.7515	12.6	19.3766	13.3	14.6993	22.0851	18.5	22.5018	19.4
8000	1	36	36	36	37.2	37.2	42.4	42.4	42.4	43.2	43.2
	2	19.3005	20.8	20.8	22.6	22.6	24.0567	27.8	27.8	28	28
	3	13.527	18	15.0629	18.6	17.4623	17.5165	21.2	21.4613	21.6	22.1278
	4	10.593	15.4305	12.2	15.9449	13.3	13.9814	18.1737	17.3	18.5166	18.3
	5	8.7525	15.003	10.08	15.5031	10.64	11.7607	17.6702	14.8	18.0036	15.52
6000	1	27	27	27	27.9	27.9	31.8	31.8	31.8	32.4	32.4
	2	14.4765	15.6	15.6	16.95	16.95	18.0465	20.85	20.85	21	21
	3	10.1475	13.5	11.3	13.95	13.1	13.1387	15.9	16.1	16.2	16.6
	4	7.9456	11.574	9.15	11.9598	9.975	10.4876	13.6316	12.975	13.8888	13.725
	5	6.5676	11.25	7.56	11.625	7.98	8.8195	13.25	11.1	13.5	11.64
4000	1	18	18	18	18.6	18.6	21.2	21.2	21.2	21.6	21.6
	2	9.6525	10.4	10.4	11.3	11.3	12.031	13.9	13.9	14	14
	3	6.7652	9	7.5315	9.3	8.7311	8.7609	10.6	10.7307	10.8	11.0639
	4	5.2965	7.7175	6.1	7.9748	6.65	6.9917	9.0895	8.65	9.261	9.15
	5	4.3785	7.5015	5.04	7.7515	5.32	5.883	8.8351	7.4	9.0018	7.76
2000	1	9	9	9	9.3	9.3	10.6	10.6	10.6	10.8	10.8
	2	4.8285	5.2	5.2	5.65	5.65	6.0155	6.95	6.95	7	7
	3	3.384	4.5	3.7629	4.65	3.7629	4.3831	5.3	5.3613	5.4	5.5278
	4	2.6505	3.861	3.05	3.9897	3.325	3.498	4.5474	4.325	4.6332	4.575
	5	2.1915	3.753	2.52	3.8781	2.66	2.9463	4.4202	3.7	4.5036	3.88

where all such paths were not edge-disjoint. Comparison of simulated routing time between all the five routing algorithms *LCMR*, *FMLBRT* [11], *MAODVRT* [13], *FMLBHC* [12] and *MAODVHC* [13] are shown in Tables 2 and 3.

In both Tables 2 and 3, the first column indicates the path numbers of k paths, the second column shows the initial estimates of routing time along these k-shortest paths between a given source-destination pair. The next three columns, each consisting of three sub-columns, show the comparison between three routing algorithms *LCMR*, *FMLBRT* [11] and *MAODVRT* [13]. The next column gives the routing time taken by k-shortest paths (between same source-destination pair) selected on the basis of hop counts along the corresponding path. The last two columns, each consisting of three sub-columns, represent the comparison between two routing algorithms *FMLBHC* [12] and *MAODVHC* [13]. In Table 3, actual routing time in the simulation turns out to be always higher than the theoretically estimated values (based on the assumption of node-disjoint paths) in each of these routing protocols due to increased queuing delay at the nodes where edges are shared. From both Tables 2 and 3, we see, however, that the *LCMR* protocol takes the minimum time for each value of k, the number of multiple paths.

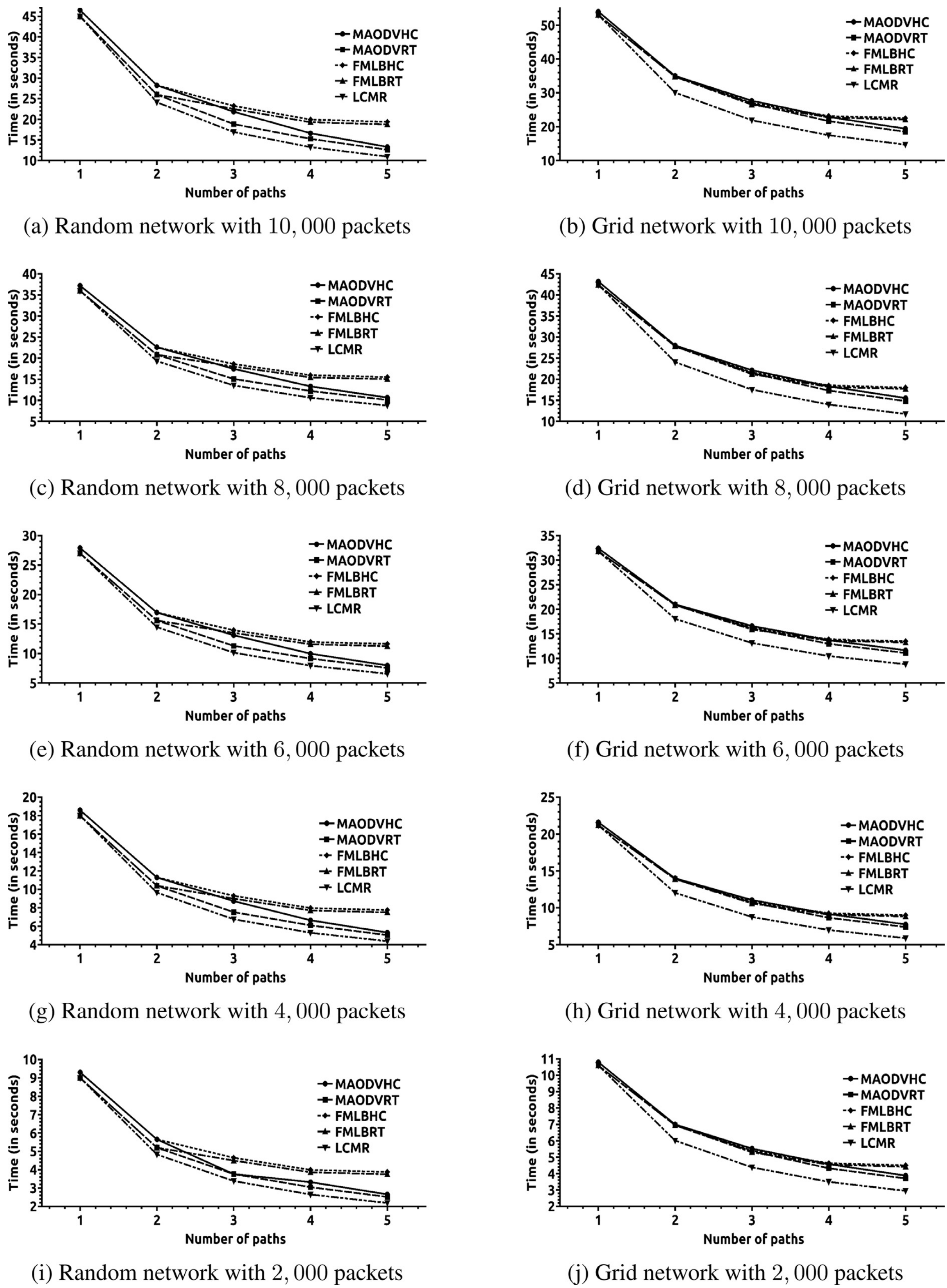
Next we performed the simulation with randomly chosen multiple source-destination pairs with traffic arrival pattern following Poisson distribution as mentioned above. Considering a large number of such cases in the simulation, we compute the average time taken by the proposed *LCMR* protocol, *FMLBRT*, *FLMBHC*, *MAODVRT* and *MAODVHC* protocols in the dynamic situation. Table 4 shows the average time taken in transmitting 10,000, 8,000, 6,000, 4,000 and 2,000 packets from a source to a destination in random networks and grid networks through 5, 4, 3, 2 and 1 multipaths, respectively.

Fig. 2(a), (c), (e), (g) and (i) show the average time taken in transmitting 10,000, 8,000, 6,000, 4,000 and 2,000 packets from a source to a destination in random networks through 5, 4, 3, 2 and 1 multipaths, respectively. Similarly, Fig. 2(b), (d), (f), (h) and (j) show the average time taken in transmitting 10,000, 8,000, 6,000, 4,000 and 2,000 packets from a source to a destination in regular grid networks through 5, 4, 3, 2 and 1 multipaths, respectively.

Also, Fig. 3(a), (c), (e) and (g) show the nature of variation of routing time against the number of packets transmitted from a source to a destination in random networks for transmission along 5, 4, 3, 2 and 1 multipaths, respectively. Similarly, Fig. 3(b), (d), (f) and (h) show similar plots for the regular grid networks corresponding to 5, 4, 3, 2 and 1 multipaths, respectively. From Figs. 2 and 3, we see that the time taken by our protocol is always much less than that taken by any of the *FMLBRT*, *FLMBHC*, *MAODVRT* and *MAODVHC* for sending data and this difference is more pronounced when the number of multipaths increases.

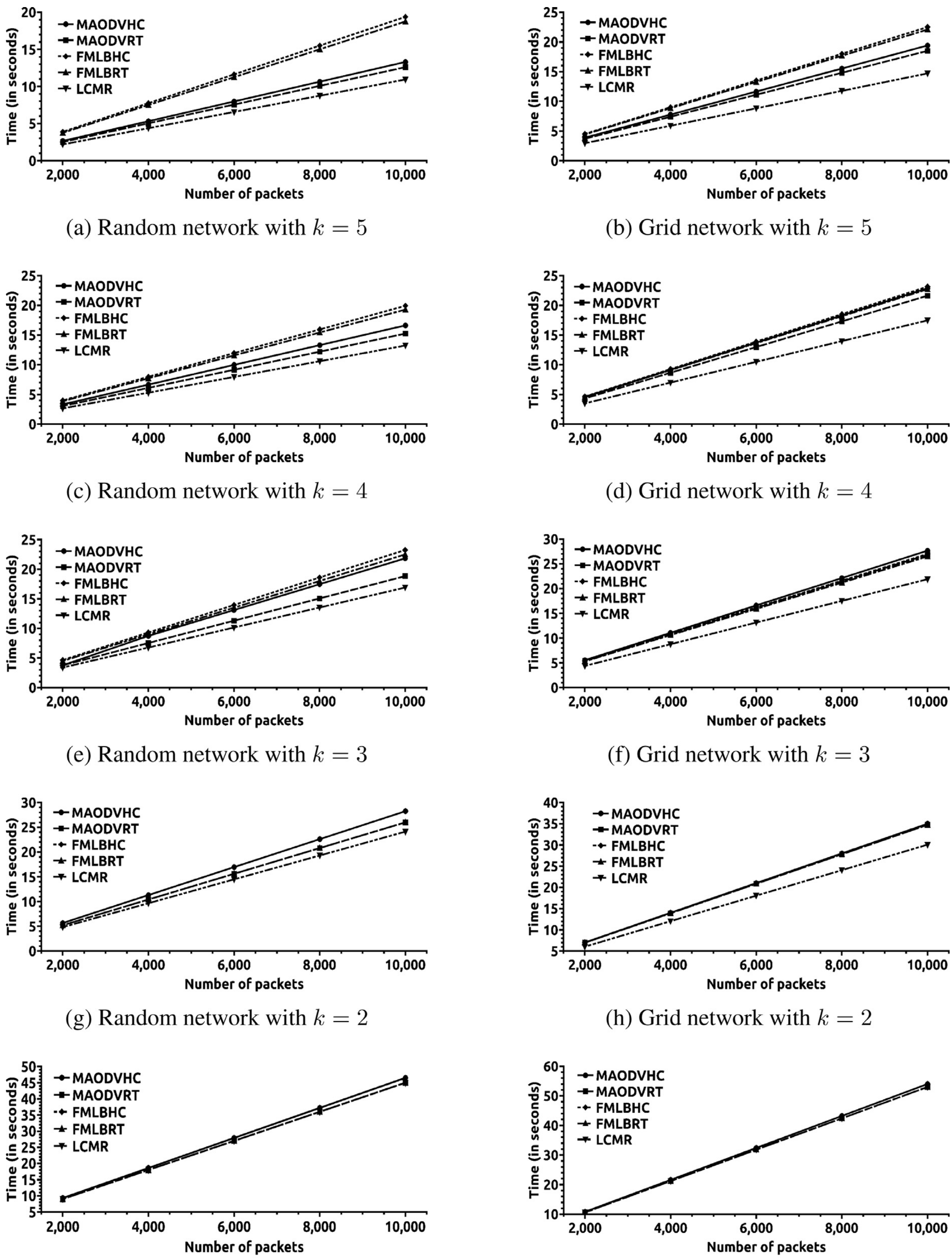
5. Conclusion

We have proposed a new load-balanced protocol called *Least Common Multiple based Routing* (*LCMR*) protocol for multipath routing in *MANET*s. With *LCMR*, while distributing the packets along multiple paths, the routing time along different paths are properly used for computing the number of data packets to be routed along each path so that the overall routing time is minimized. This is in contrast to the policy used in *FMLB* protocols (based on both hop count [12] and routing time [11]) where the number of data packets routed along different paths are in the ratio of Fibonacci numbers which disregards the exact variation in routing time along these paths. The performance improvement of our proposed *LCMR* protocol in respect of routing time over the *FMLB* and *MAODV* protocols have been shown by a theoretical analysis as well as through simulation studies.



(a) Random network with 10,000 packets
(b) Grid network with 10,000 packets
(c) Random network with 8,000 packets
(d) Grid network with 8,000 packets
(e) Random network with 6,000 packets
(f) Grid network with 6,000 packets
(g) Random network with 4,000 packets
(h) Grid network with 4,000 packets
(i) Random network with 2,000 packets
(j) Grid network with 2,000 packets

Fig. 2. Average routing time (end-to-end delay) vs. number of paths.



(a) Random network with $k = 5$

(b) Grid network with $k = 5$

(c) Random network with $k = 4$

(d) Grid network with $k = 4$

(e) Random network with $k = 3$

(f) Grid network with $k = 3$

(g) Random network with $k = 2$

(h) Grid network with $k = 2$

(i) Random network with $k = 1$

(j) Grid network with $k = 1$

Fig. 3. Average routing time (end-to-end delay) vs. number of packets.

References

[1] C.E. Perkins, P. Bhagwat, Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers, SIGCOMM Comput. Commun. Rev. 24 (4) (1994) 234–244.

[2] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, L. Viennot, Optimized link state routing protocol for ad hoc networks, in: Proceedings of the IEEE International Multi Topic Conference Technology for the 21st Century, 2001, pp. 62–68.

[3] C.E. Perkins, E.M. Royer, Ad-hoc on-demand distance vector routing, in: Proceedings of the Second IEEE Workshop on Mobile Computing Systems and Applications, 1999, pp. 90–100.

[4] S.K. Ghosh, K. Sinha, On convex greedy embedding conjecture for 3-connected planar graphs, in: Proc. 17th Intl. Symp. on Fundamentals of Computation Theory (FCT), Poland, 2009, pp. 145–156.

[5] V.D. Park, M.S. Corson, A performance comparison of the temporally-ordered routing algorithm and ideal link-state routing, Computers and Communications, 1998. ISCC'98, in: Proceedings. Third IEEE Symposium on, Athens, 1998, pp. 592–598.

[6] D.B. Johnson, D.A. Maltz, Truly seamless wireless and mobile host networking. protocols for adaptive wireless and mobile networking, IEEE Pers. Commun. 3 (1) (1996) 34–42.

[7] M.R. Pearlman, Z.J. Haas, Determining the optimal configuration for the zone routing protocol, IEEE J. Sel. Areas Commun. 17 (8) (1999) 1395–1414.

[8] W.R. Heinzelman, A. Chandrakasan, H. Balakrishnan, Energy-efficient communication protocol for wireless microsensor networks, in: Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, 2, 2000, p. 10.

[9] S. Lindsey, C. Raghavendra, Pegasis: power-efficient gathering in sensor information systems, in: Proceedings of the IEEE Aerospace Conference, 3, 2002, pp. 1125–1130.

[10] D.A. Tran, H. Raghavendra, Routing with congestion awareness and adaptivity in mobile ad hoc networks, in: IEEE Wireless Communications and Networking Conference, 4, 2005, pp. 1988–1994.

[11] M. Naseem, C. Kumar, Congestion-aware Fibonacci sequence based multipath load balancing routing protocol for manets, Wireless Pers. Commun. 84 (4) (2015) 2955–2974.

[12] Y. Tashtoush, O. Darwish, M. Hayajneh, Fibonacci sequence based multipath load balancing approach for mobile ad hoc networks, Ad Hoc Netw. 16 (2014) 237–246.

[13] Y. Ganjali, A. Keshavarzian, Load balancing in ad hoc networks: single-path routing vs. multi-path routing, in: INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, 2, 2004, pp. 1120–1125 vol.2, doi:10.1109/INFCOM.2004.1356998.

[14] M.R. Pearlman, Z.J. Haas, P. Sholander, S.S. Tabrizi, On the impact of alternate path routing for load balancing in mobile ad hoc networks, in: Mobile and Ad Hoc Networking and Computing, 2000. MobiHOC. 2000 First Annual Workshop on, 2000, pp. 3–10, doi:10.1109/MOBHOC.2000.869207.

[15] L. Wang, Y. Shu, M. Dong, L. Zhang, O.W.W. Yang, Adaptive multipath source routing in ad hoc networks, in: Communications, 2001. ICC 2001. IEEE International Conference on, 3, 2001, pp. 867–871 vol.3, doi:10.1109/ICC.2001.937362.

[16] G. Parissidis, V. Lenders, M. May, B. Plattner, Multi-path Routing Protocols in Wireless Mobile Ad Hoc Networks: A Quantitative Comparison, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 313–326. doi:10.1007/11759355_30.

[17] P.P. Pham, S. Perreau, Performance analysis of reactive shortest path and multipath routing mechanism with load balance, in: INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies, 1, 2003, pp. 251–259 vol.1, doi:10.1109/INFCOM.2003.1208677.

[18] A. Nasipuri, R. Castañeda, S.R. Das, Performance of multipath routing for on-demand protocols in mobile ad hoc networks, Mobile Netw. Appl. 6 (4) (2001) 339–349, doi:10.1023/A:1011426611520.

[19] T. Gonzalez, O. Ibarra, S. Sahni, Bounds for LPT schedules on uniform processors, SIAM J. Comput. 5 (1) (1977) 155–166.

[20] T. Gonzalez, S. Sahni, Preemptive scheduling of uniform processor systems, J. ACM 25 (1978) 92–101.

[21] K. Sinha, A.D. Chowdhury, S. Banerjee, S.K. Ghosh, Efficient load balancing on a cluster for large scale online video surveillance, in: Proc. 10th Intl. Conf. on Distributed Computing and Networking (ICDCN), LNCS 5408, Hyderabad, India, 2009, pp. 450–455.

Ansuman Bhattacahrya obtained his B. Tech. degree in Electronics and Communication Engineering from West Bengal University of Technology, Kolkata in 2006 and M. Tech. degree with specialization in Radiophysics and Electronics from University of Calcutta, Kolkata in 2008. In 2016, he obtained his PhD from Indian Statistical Institute, Kolkata. From October 2008 to January 2009, he joined Tech Mahindra as a Technical Associate. From March 2009 to July 2009, he worked as a Lecturer in Guru Nanak Institute of Technology. In July 2009, he joined the Indian Statistical Institute, Kolkata as a Project Linked Personnel and since April 2011 to July 2014 he worked in the Advanced Computing and Microelectronics Unit as a Senior Research Fellow. Now, He has been working as an Assistant Professor at Dept. of Computer Science and Engineering in N. I. T. Meghalaya from August 2014. His research interests include channel allocation and routing in cognitive radio networks, secure wireless communication and energy-efficient communication. He is a Member of the IEEE.

Koushik Sinha is an Assistant Professor in the Department of Computer Science, Southern Illinois University, Carbondale. Prior to that, he was with the Social Computing Group of Qatar Computing Research Institute. Previously, he was a Research Scientist at Hewlett-Packard Labs. He also spent 7 years with Honeywell as a Lead Research Scientist. He has 7 granted patents, is the author of over 50 publications, and a book. Koushik received the Young Scientist Award from the Indian Science Congress in 2009. His current research areas are social computing, wireless sensor networks, and P2P networks. He is a Senior Member of the IEEE.