A Novel Delay-Oriented Shortest Path Routing Protocol for Mobile Ad Hoc Networks

Shiann-Tsong Sheu[+] and Jenhui Chen[*]

[+]Department of Electrical Engineering
*Department of Computer Science and Information Engineering
Tamkang University, Tamsui, Taipei, Taiwan 25137, R.O.C.
E-mail: stsheu@ee.tku.edu.tw

***Abstract*– In wireless *ad hoc* mobile network, a host desires to communicate with another host may need some intermediate nodes to relay data packets. To maximize the channel resource utilization and minimize the network transfer delay along the path, the shortest path with minimum hops approach is often adapted. However, by considering the employing medium access control (MAC) protocol, the minimum transfer delay from source to destination may be achieved by choosing a longer path but with less contention delay. In this paper, we will propose an efficient delay-oriented routing protocol for mobile *ad hoc* wireless networks. The expected access contention delay of IEEE 802.11 protocol is analyzed to support the routing decision. Simulation results show that the derived path length in proposed delay-oriented routing protocol is slightly higher than that of conventional shortest path with minimum hops approach but it can significantly reduce both average transfer delay and packet loss rate.**

I. INTRODUCTION

A wireless *ad hoc* network is a collection of mobile hosts, which form a temporary network without the aid of any pre-established infrastructure or centralized administration. When the network population is large, the set of nodes is often partitioned into clusters so that the resource can be handled in an efficient way. Generally, a cluster is defined as a number of mobile hosts, which can directly transmit/receive packets to/from each other and content the same network bandwidth. Mobile members in a cluster are often located within a limited coverage area, which is decided by the transmission power. Moreover, a mobile host is allowed to belong to many clusters at any time. Since all members of a cluster share the channel resource, member in a 'bigger' cluster will have a higher probability of suffering a longer medium access control (MAC) delay.

The most important issue in a wireless *ad hoc* network is how a mobile host to communicate with another mobile host, which is not in its direct transmission range. Intuitively, the transmitted packets from source must be relayed via some intermediate hosts. The critical problem is how to find an efficient and reliable route from source to destination. The common approach is to consider the shortest-path routing. The well-known algorithm is the Distributed Bellman-Ford algorithm (DBF) [1]. In DBF, every host in the network maintains the length (cost) of the shortest path from each of its neighbor hosts to every destination in the network. With this information, a host sends data packets to a neighbor, which leads to a shortest path to the destination. In order to maintain up-to-date distance information in a dynamic environment, every host monitors its outgoing link and periodically broadcasts to neighboring hosts its current estimation of the shortest distance to every network destination.

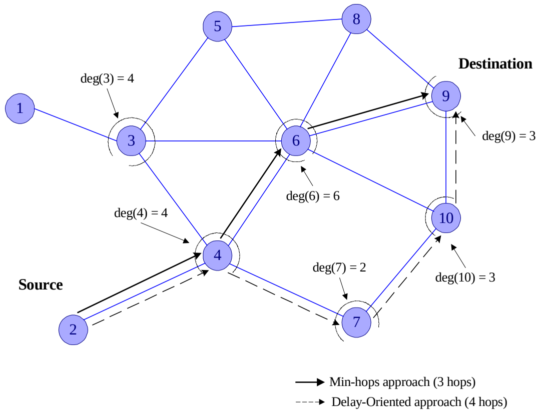


Fig. 1. An example of routing in wireless *ad hoc* network.

The most commonly used measurement of distance is the number of hops in the path. Even though this measure is easy to compute, it cannot reflect the influences on realistic access delays. That is, packets follow the shortest path with minimum hop count may take a considerable time to reach destination. This is because that a routing algorithm, which is based on such a distance measurement, may route almost packets over a few (shortest-distance) paths in network. Each time the selected intermediate node relaying the packets needs a longer access and contention delay. This will result in serious congestion in network, especially in the wireless network with scare bandwidth. Taking Fig. 1 for example, if source STA 2 wants to send packets to STA 9, the shortest path with the minimum hop will be the path [v_2, v_4, v_6, v_9]. Along this path, when STA 6 relays packets, it needs to contend the air channel with the other 6 stations STA 3, 4, 5, 8, 9 and 10. This may spend a long time to solve the channel contention by any contention-based protocol. Accordingly, the MAC delay will become very larger if the routing algorithm keeps routing other packets to pass through hot spot STA 6. On the contrary, if we select the path [v_2, v_4, v_7, v_{10}, v_9] with 4 hop counts, the relayed packets have a better chance to quickly reach destination. Therefore, it is desired to design an efficient delay-oriented shortest path routing (DOSPR) protocol for wireless *ad hoc* networks. In this paper, we will propose a DOSPR protocol for the IEEE 802.11 Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) wireless *ad hoc* network with moderate mobility scenarios.

The reminder of this paper is organized as follows. Section II will describe the proposed DOSPR protocol in detail.

Moreover, the way of predicting the medium access delay in IEEE 802.11 CSMA/CA wireless *ad hoc* network is analyzed. The simulation environments and results are shown in Section III. Section IV presents the conclusion remarks.

II. The DOSPR Protocol

In this section, we will present the DOSPR protocol. Before describing the DOSPR protocol, two critical problems must be solved: (1) In order to find the ‘best’ route with minimum access delay, the DOSPR protocol needs collect all network information on time. (2) The DOSPR needs to predict the precise medium access delay of a node in IEEE 802.11 wireless networks.

Employing some well-known on-demand routing protocols, for instances, the Dynamic Source Routing (DSR) [2] and the Ad-Hoc On-demand Distance Vector (AODV) [3] routing protocols, can solve the first problem. In these protocols, the routes are established on data transmission demand by a source host. In the DSR algorithm, the source host determines the complete sequence of hosts in the routing path. In wireless network, since the network connectivity is changing from time to time, the source routes are dynamically constructed using a *route-discovery* protocol, i.e., whenever a host needs a route to another host and it does not have one in its cache, it dynamically determines one by flooding the network with route-discovery packets. Another approach is by using table-driven algorithm; each router/host maintains information for each known destination in the network and updates its routing-table entries as needed. Examples of table driven based on distance vectors are the routing protocols of the DSDV [4] and WRP [5]. For the sake of simplicity, our DOSPR employs the table-driven approach.

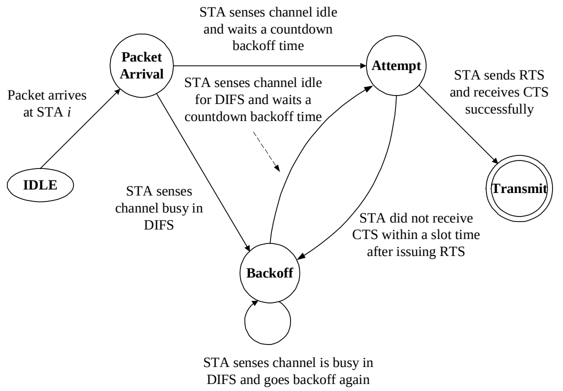
The second problem can be solved by we analyze the access delay in the CSMA/CA protocol. In order to calculate the access delay and find the available path, each station needs maintain a *connective status matrix* (*CSM*) to record the connective status in the network. The *CSM* is defined as follows.

- *Connective Status Matrix*: $CSM = \{s(u,v)_{N\times N} \mid 1\leq u,v\leq N\}$, where $s(u,v)=k$, $k\in\{0,1\}$. Element $s(u,v)=k$ $(k>0)$ indicates that vertex u can transmit packets to vertex v directly. Otherwise, vertices u and v cannot hear each other.

For illustration, consider the example shown in Fig. 1 again. The derived *CSM* matrix is shown as follows.

$$CSM = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

The matrix of this example is symmetry. However, in real world, the transmission condition between two hosts may not be the same in both directions. This implies that the symmetry feature is not necessary for *CSM*. According to the *CSM*, every source node can apply the Dijkstra algorithm to find the shortest path with the minimum hop count [6] to the desired destination. (We note that value 0 in *CSM* matrix should be treated as infinite positive value when applying the Dijkstra algorithm.) In this paper, the proposed DOSPR protocol is similar to the Dijkstra’s algorithm excepting the cost function on edges. To obtain the path with the minimum access delay, we need modify the value of each element in the *CSM* matrix as the desired cost value, which is the predicted access delay. Now we will describe how to decide the delay cost $s(u,v)$ of node u transmitting packet to node v.



STA senses channel idle and waits a countdown backoff time
Packet Arrival
Attempt
Packet arrives at STA *i*
STA senses channel idle for DIFS and waits a countdown backoff time
STA sends RTS and receives CTS successfully
IDLE
Transmit
STA senses channel busy in DIFS
STA did not receive CTS within a slot time after issuing RTS
Backoff
STA senses channel is busy in DIFS and goes backoff again

Fig. 2. The transition state diagram of the DOSPR on STA *i*.

A. Delay Cost Estimation

Recall the *Distributed Coordination Function* (DCF) of IEEE 802.11 [7] is used as the MAC protocol to avoid the collision. It uses *Request-To-Send* (RTS) and *Clear-To-Send* (CTS) control packets to overcome the well-known hidden terminal problem and to provide virtual carrier sense for saving battery power. In this paper, we assume each data transmission should first issue RTS and CTS, and follow by an *acknowledgment* (ACK). The DCF needs two basic inter-frame spaces (*DCF Inter-Frame Space* (DIFS) and *Short Inter-Frame Space* (SIFS)) for supporting asynchronous data transmission. The SIFS is used to guarantee the control packets to have a higher priority than data packets. Besides, each time a station wants to transmit data packet must sense channel idle at least for DIFS time interval. Therefore, the SIFS is shorter than DIFS.

Fig. 2 illustrates the simplified transition state diagram of STA *i* attempts to transmit packets in IEEE 802.11 standard. Initially, STA *i* stays in *IDLE* state. When packet arrives STA *i* (either generated by itself or received by neighbor for relaying), STA *i* will enter into *Packet_Arrival* state. In this state, if STA *i* senses medium busy in SIFS period, it recognizes the channel is busy and enters the *Backoff* state right away. Otherwise, if the channel sustains idle for DIFS period, it will enter the *Attempt* state and delay a random *backoff* time interval (denoted as $\tilde{b}$) before transmission.

For simplicity, we let $P_{idle}^{i}(t)$ denote the probability of STA *i* successes in sensing channel idle for time interval *t*.

(Also, the $P_{idle}^{i}(t)$ can be treated as the probability that STA *i* detects no other station transmitting data during observing time interval *t*.) Therefore, the probabilities of the state transition from state *Packet_Arrival* to states *Attempt* and *Backoff* are $P_{idle}^{i}(DIFS)$ and 1-$P_{idle}^{i}(DIFS)$ respectively. The required delays for the former and latter state transitions are SIFS and DIFS+$\tilde{b}$ respectively. In the *Backoff* state, STA *i* has the probability 1-$P_{idle}^{i}(DIFS)$ to sense channel busy after finishing its countdown. In this case, it will delay RTS+SIFS+CTS+SIFS+packet_len+ SIFS+ACK before its next attempt (Here, we uses notations RTS/CTS/ACK and packet_len for the required time periods of transmitting a RTS/CTS/ACK control packet and a data packet, respectively).

Once STA *i* detects channel idle (with probability $P_{idle}^{i}(DIFS)$), it will enter *Attempt* state to transmit packet. In the *Attempt* state, STA *i* will first issue the RTS control packet and then waits for the CTS packet to make sure the contention is success. If no CTS is detected within a slot_time (the slot_time is defined as the time unit in the backoff process), STA *i* will return *Backoff* state immediately. The probability of occurring collision (i.e., failing on receiving CTS) is 1-$P_{idle}^{i}(slot)$ and the waste time is RTS + 2×SIFS. On the contrary, STA *i* has the probability $P_{idle}^{i}(slot)$ to transmit packet in success. In this case, it needs RTS+SIFS+CTS+ SIFS time period to make sure the reservation is success.

Now, we will calculate the probability $P_{idle}^{i}(t)$ and the average backoff time $\tilde{b}$. Assume the packet arrival rate of a mobile station follows the Poisson distribution and the average arrival rate of a station is λ. Let $P_n(t)$ denotes the probability of *n* packets arrive a station during interval time *t*. We have

$$P_n(t) = \frac{(\lambda t)^n}{n!} e^{-\lambda t}$$

Hence, the probability of no packet arrive at station during the interval time *t* is

$$P_0(t) = e^{-\lambda t}$$

In the IEEE 802.11 CSMA/CA wireless network, stations in a cluster will contend and share the bandwidth. For simplicity, we join the Poisson processes of multiple sources as an aggregate Poisson process. Let | *Adj*(*i*) | be the number of neighbor stations of STA *i*. According to *CSM*, the |*Adj*(*i*)| can be easily derived by the following equation:

$$|\,Adj(i)\,| = \sum_{v=1}^{N} s(i,v).$$

From the point of view of STA *i*, the total packet arrival rate of other stations in cluster is $\hat{\lambda}$ = | *Adj*(*i*) | × λ. Therefore, the probability $P_{idle}^{i}(t)$ can be derived as follows:

$$P_{idle}^{i}(t) = e^{-\hat{\lambda} t}.$$

Now we can estimate the expected delays encountered in the *Attempt* State (*EA*(*i*)) and *Backoff* State (*EB*(*i*)). Thus, we have

$$\begin{aligned} EA(i) &= P_{idle}^{i}(slot) \times (RTS + 2 \cdot SIFS + CTS) \\ &\quad + (1 - P_{idle}^{i}(slot)) \times (RTS + 2 \cdot SIFS + EB(i)) \end{aligned}$$

and

$$\begin{aligned} EB(i) &= P_{idle}^{i}(DIFS) \times (DIFS + \tilde{b} + EA(i)) \\ &\quad + (1 - P_{idle}^{i}(DIFS)) \times (\overline{B} + EB(i)). \end{aligned}$$

where $\overline{B} = RTS + 3 \cdot SIFS + CTS + packet\_len + ACK$.

To simplify the *EB*(*i*), we derive

$$\begin{aligned} EB(i) &= \frac{1}{P_{idle}^{i}(DIFS) \times P_{idle}^{i}(slot)} \times \\ &\quad [P_{idle}^{i}(DIFS) \times (DIFS + \tilde{b} + RTS + 2 \cdot SIFS + P_{idle}^{i}(slot) \times CTS) \\ &\quad + (1 - P_{idle}^{i}(DIFS)) \times \overline{B}]. \end{aligned}$$

Now we will solve the parameter $\tilde{b}$ in equation *EB*(*i*). Recall symbol $\tilde{b}$ is the mean back-off time of transmission. Let *W* denote the specified contention window size. In this paper, we assume *W*=32 time slots and the maximum window size for retransmission is 1024 time slots. According to the binary exponential backoff algorithm in CSMA/CA protocol, the backoff delay *b*(*n*) of the *n*-th retransmission (0≤*n*≤5) can be calculated by the following recursive function:

$$\begin{aligned} b(0) &= P_{idle}^{i}(slot) \times \frac{2^0 \cdot W}{2} + (1 - P_{idle}^{i}(slot)) \times b(1) \\ b(1) &= P_{idle}^{i}(slot) \times \frac{2^1 \cdot W}{2} + (1 - P_{idle}^{i}(slot)) \times b(2) \\ b(2) &= P_{idle}^{i}(slot) \times \frac{2^2 \cdot W}{2} + (1 - P_{idle}^{i}(slot)) \times b(3) \\ b(3) &= P_{idle}^{i}(slot) \times \frac{2^3 \cdot W}{2} + (1 - P_{idle}^{i}(slot)) \times b(4) \\ b(4) &= P_{idle}^{i}(slot) \times \frac{2^4 \cdot W}{2} + (1 - P_{idle}^{i}(slot)) \times b(5) \\ b(5) &= \frac{2^5 \cdot W}{2} = 2^4 \cdot W. \end{aligned}$$

Then, we obtain

$$\begin{aligned} \tilde{b} &= \sum_{n=0}^{4} \left(P_{idle}^{i}(slot) \times (1 - P_{idle}^{i}(slot))^n \times 2^{n-1} \times W \right) \\ &\quad + (1 - P_{idle}^{i}(slot))^5 \times 2^4 \times W. \end{aligned}$$

Finally, we can get the delay cost (including contention and transmission delay) of the STA *i* as follows:

$$\begin{aligned} D_{delay}^{i} &= P_{idle}^{i}(DIFS) \times (DIFS + \tilde{b} + EA(i)) \\ &\quad + (1 - P_{idle}^{i}(DIFS)) \times (SIFS + EB(i)) \\ &\quad + \frac{packet\_len}{data\_tate}. \end{aligned}$$

Fig. 3 shows the expected MAC delay of STA *i* under different number of neighbor nodes in a cluster when the packet arrival rate λ is 0.1 and the packet mean length is 20 time slots (a slot time is 20 μs) and the data rate is 2Mbits/sec. It is clear that the MAC delay is proportional with the number of competitors. We also notice that, we do not consider the buffer delay for the delay cost in this paper. The reason is that the precise buffer delay is very hard to be obtained from mobile users. Fortunately, the MAC contention delay can be roughly treated as the buffer delay. The reason is a smaller MAC delay may result in the buffered packets can be quickly serviced. Thus, we only use the MAC and transmission delay as the delay cost in the DOSPR.

Based on the derived cost delay of STA *i*, we replace every non-zero element in the *i*-th column in *CSM* by D_{delay}^{i}. (That is, $s(i,j)= s(i,j)\times D_{delay}^{i}$, $\forall 1\le j\le N$.) The shortest path of the minimal delay can be found by employing the Dijkstra algorithm [7]. (We also note that each element of zero indicates infinite delay cost in Dijkstra algorithm).

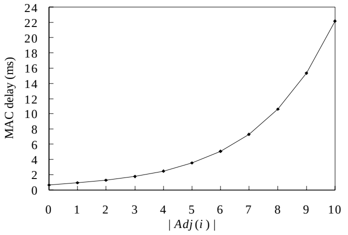


Fig. 3. The expected MAC delay of STA *i* under different number of neighbor nodes.

Let's consider the network shown in Fig. 1 again and use the same assumption with λ = 0.1 and packet mean length is 20 then the final *CSM* matrix for DOSPR will become

$$CSM = \begin{bmatrix} 0 & 0 & 2.2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2.2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1.0 & 0 & 0 & 2.2 & 1.6 & 4.2 & 0 & 0 & 0 & 0 \\ 0 & 1.0 & 2.2 & 0 & 0 & 4.2 & 1.3 & 0 & 0 & 0 \\ 0 & 0 & 2.2 & 0 & 0 & 4.2 & 0 & 1.6 & 0 & 0 \\ 0 & 0 & 2.2 & 2.2 & 1.6 & 0 & 0 & 1.6 & 1.6 & 1.6 \\ 0 & 0 & 0 & 2.2 & 0 & 0 & 0 & 0 & 0 & 1.6 \\ 0 & 0 & 0 & 0 & 1.6 & 4.2 & 0 & 0 & 1.6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4.2 & 0 & 1.6 & 0 & 1.6 \\ 0 & 0 & 0 & 0 & 0 & 4.2 & 1.3 & 0 & 1.6 & 0 \end{bmatrix}$$

According to the conventional shortest path of min-hop counts, the path [v_2, v_4, v_6, v_9] will take 2.2 + 4.2 + 1.6 = 8.0 (ms) for every packet to reach destination. On the contrary, using the path [v_2, v_4, v_7, v_{10}, v_9] for route will lead lower delay 2.2 + 1.3 + 1.6 + 1.6 = 6.7 (ms). It is apparent that the second path with more hop counts [v_2, v_4, v_7, v_{10}, v_9] will get lower delay by 1.3 (ms). Let's consider another case in this example, the source is STA 2 and the destination is STA 10. The shortest path of min-hop count approach can be either the path [v_2, v_4, v_6, v_{10}] or path [v_2, v_4, v_7, v_{10}]. We can see that these two paths have the same hop counts but they will lead to different delays. Obviously, the path <v_2, v_4, v_7, v_{10}> with transfer delay 5.1 ms is better than the path <v_2, v_4, v_6, v_{10}> with total delay 8.0 ms. This is because STA 6 is the bottleneck for relaying packets.

III. Simulation Model and Results

To evaluate the effectiveness of the proposed DOSPR protocol is implemented by C++ programming language. The IEEE 802.11 Medium Access Control Protocol is employed as the Data Link Layer. In simulations, we consider the realistic system parameters, which are shown in Table I.

TABLE I.
System Parameters in Simulations

Parameter	Normal Value
Channel bit rate	2Mbps
ACK frame length	112 bits
Slot Time (slot)	20 μs
SIFS	10 μs
DIFS	50 μs
PHY and MAC header	400 bits
Transmission time of RTS	144 μs
Transmission time of CTS	120 μs

A. *Simulation Environment*

In our simulations, we simulated a scenario of 20 hosts simultaneously active in a square area of 600m x 600m. The initial location of each host is assigned randomly. Each host has a transmission range of 200m. In our simulations, we consider two different models. In the first simulation model (model I), hosts are static during whole simulation period. In the second simulation model (model II), every host is movable. The moving probability in simulation is considered from 0.1 to 1.0 in a step of 0.1. The distance of each moving is 100m and the move direction is randomly selected from 8 directions. To reflect the realistic situation, each time a station decides to move, it will stay at the new position for at least 20 seconds before its next move. In other words, the maximum moving speed of a mobile host is 5 m/s.

Each simulation run is last 200 seconds (≈ 10^7 slot times). The packet arrival rate of each mobile host follows the Poisson distribution with a mean λ, and the packet length is an exponential distribution with a mean of *L* slots. The packet mean length is according to the analyzed average network packets on ordinary LAN [8], which is about 50 Bytes ~ 150 Bytes (i.e., about 10 slots ~ 30 slots in 2Mbps transmission rate). These popular TCP/UDP packets occupy overall traffic loading over 74%. Thus, we assume *L*=20 slots in our simulations.

In order to evaluate the efficiency of proposed DOSPR protocol, we investigate two parameters: the average path length (in hop-count) and the average transfer delay. The average transfer delay is defined as the average delay, which including the MAC delay, Buffer Queuing delay and transmission delay, of a packet traveling from source to destination. For the sake of comparison, the conventional shortest path with minimum hop-count approach (denoted as Min-hops in abbreviation) is considered.

B. *Simulation Results*

Fig. 4 and 5 show the derived average transfer delays and average path lengths of DOSPR and Mini-hop approach in model I under different packet arrival rate and transmission

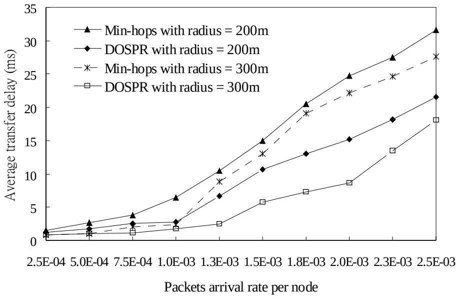


Fig. 4. Comparisons of the derived average transfer delays by DOSPR and Min-hops approach under different packet arrival rate in model I.

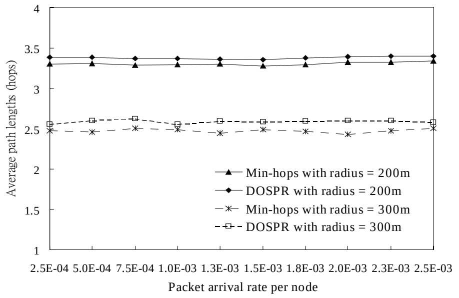


Fig. 5. Comparisons of the derived average path lengths by DOSPR and Min-hops approach under different packet arrival rate in model I.

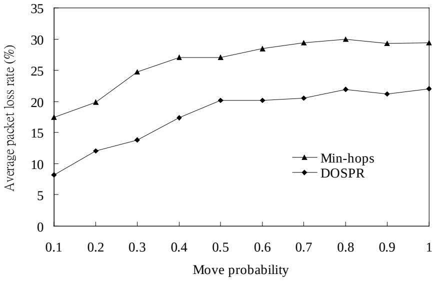


Fig. 6. Comparisons of derived average packet loss rates by DOSPR and Min-hops approach under different packet arrival rates in model II when $\lambda = 10^{-3}$ and L = 20 slots.

range (radius). A higher packet arrival rate indicates a higher network load. In Fig. 4, we can see that DOSPR provides a lower average transfer delay than Min-hop approach under the same transmission radius no matter what network load is. Moreover, we can also find that the average path length of DOSPR is only slightly higher than Min-hop approach in Fig. 5. We notice that the average transfer delay improvement is made by DOSPR reducing the MAC contention delay along the selected path. As enlarging the transmission radius to 300m (by controlling the transmission power), the path length and the average packet delay can be further reduced in both approaches. In this case, the path length difference between both approaches is still very small. Nevertheless, we can see that the transfer delay improvement in the DOSPR is more obvious than Min-hop approach. This phenomena shows that the transfer delay is somewhat dominated by the buffer queuing delay. The reason is the Min-hops approach will forward lots of packets over few hosts to minimize the path length. Therefore, even though both the transmission delay and the number of intermediate nodes are reduced, the contention delay and buffer delay occurring on a selected intermediate node may become higher than usual.

Fig. 6 shows that DOSPR protocol has the better ability to handle the network mobility. Obviously, the derived packet loss ratio by DOSPR is always smaller than that of Min-hops approach in simulation model II. Recall the DOSPR will select a longer path to obtain less transfer delay. Hence, the distance between two adjacent hosts may be less than that in the Min-hops approach. This implies that the selected hops in Min-hops approach have a higher probability located near the boundary of transmission range. Consequently, the Min-hops approach will easily suffer from path loss and need extra rerouting overhead. This is another drawback of Min-hops approach.

According to above simulation results, it is observed that the proposed DOSPR routing protocol outperforms min-hop count routing in all cases. We also concluded that using the minimum-hop routing approach might not always gain optimal delay well since it does not consider the congestion and the air radio medium contention.

IV. Conclusion and Remark

In this paper, we present a new routing scheme, delay-oriented shortest routing (DOSPR) protocol, which provides an efficient and scalable solution for mobile *ad hoc* networks. The designed DOSPR protocol considers the access delay affections along the path. Simulation results demonstrated that the derived hop counts is slightly higher than the minimum hop counts. However, based on the proposed DOSPR protocol, the total transfer delay from source to destination of each packet can be significantly minimized. Furthermore, the packet loss ratio, which is caused by mobility, can be also reduced by the DOSPR.

References

[1] S. Murthy and J. J. Garcia-Luna-Aceves, “A Routing Protocol for Packet Radio Networks,” in *Proceedings of 1st Annual ACM International Conference on Mobile Computing and Networking*, pp. 86-95, 1995.

[2] D.B. Johnson and D.A. Maltz, ”Dynamic Source Routing in Ad Hoc Wireless Networks,” *Mobile Computing*, edited by T. Imielinski and H. Korth, Chapter 5, Kluwer Publishing Company, pp. 153-181, 1996.

[3] C.E. Perkins and E.M. Royer, “Ad-Hoc On-Demand Distance Vector Routing,” in *Proceedings of IEEE WMCSA’99*, New Orleans, LA, pp. 90-100. Feb. 1999.

[4] C. E. Perkins and P. Bhagwat, “Highly Dynamic Destination-Sequence Distance-Vector routing (DSDV) for Mobile Computers,” in *Proc. ACM SIGCOMM 94*, London, UK, October 1994.

[5] S. Murthy and J.J. Garcia-Luna-Aceves, “An Efficient Routing Protocol for Wireless Networks,” *Journal of Special Issue in Mobile Networks and Applications (MONET)*, Vol.1, no.2, pp. 183-97, October 1996.

[6] Mikhail J. Atallah, *Algorithms and Theory of Computation Handbook*. CRC Press, contents 6-13, 1999.

[7] LAN MAN Standards Committee of the IEEE Computer Society, “Wireless LAN Medium ACCESS Control (MAC) and Physical Layer (PHY) Specifications,” ANSI/IEEE Std. 802.11, 1999.

[8] K.M. Khalil, K.Q. Luc and D.V. Wilson, “LAN Traffic Analysis and Workload Characterization,” In *Proceedings of 15th Conference on Local Computer Networks*, pp. 112-122, 1990.