PAPER *Special Issue on New Developments on QoS Technologies for Information Networks*

Delay-Oriented Routing Protocol for Wireless Ad Hoc Networks

Shiann-Tsong SHEU[†], **Tihjia TSAI**[††], ***and*** **Jenhui CHEN**[††], ***Nonmembers***

SUMMARY 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.

key words: *ad hoc, MAC, QoS*

1. 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 hosts 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 if any. 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* (DBF) algorithm [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 links and periodically broadcasts to neighboring hosts its current estimation of the shortest distance to every network destination.

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 large 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.

Manuscript received November 8, 2000.

Manuscript revised January 15, 2001.

[†] The author is with the Department of Electrical Engineering, Tamkang University, Tamsui, Taipei, Taiwan 25137, Republic of China.

[††] The authors are with the Department of Computer Science and Information Engineering, Tamkang University, Tamsui, Taipei, Taiwan 25137, Republic of China.

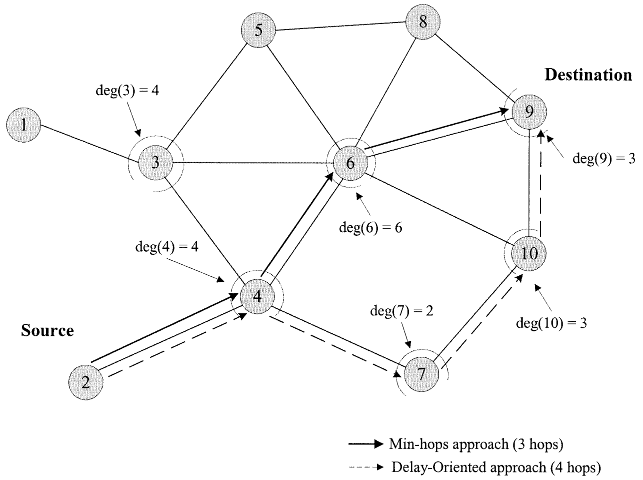

Fig. 1 An example of routing in wireless *ad hoc* network.

Section 2 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 Sect. 3. Section 4 presents the conclusion remarks.

2. 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, one may use a route-discovery protocol to dynamically construct the source routes. That is, 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.

In this paper, we use a hybrid approach (with both table-driven and on-demand routing) to collect network information and to make the routing decision. Since the information of entries of routing table may be expired, we adopt the aging function for each entry. For each new request, this hybrid approach will first find the best path from routing table. If the found path includes any expired entry, this hybrid approach will find the actual one on-demand by issuing route-discovery packets. As soon as a new path is collected, every entry in the routing table along this path will be updated and its associated timeout timer is reset. This may prevent from wasting bandwidth.

If a source node moves, it is able to reinitiate the route discovery protocol to find a new route to the destination. If a node along the route moves, its upstream neighbor, which notices the move, will update its routing table. Meanwhile, it can issue the route-discovery packet to discover the new route to the moved host. Then the intermediate host renews the entry in its routing table and notifies its upstream neighbor with this new update information, and so on until the source node is informed.

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

- *Connection Status Matrix:* $CSM = \{s(u,v)_{N\times N} \mid 1 \le u,v \le N,\ where\ s(u,v) = k,\ k_i \in \{0,1\}\}$. Element $s(u,v) = 1$ indicates that vertex u can transmit packets to vertex v directly. Otherwise, vertices u and v cannot hear with 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 [4] 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.

2.1 Delay Cost Estimation

Recall the *distributed coordination function* (DCF) of IEEE 802.11 [5] 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.

Figure 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 $P^i_{idle}(t)$ denote the probability of STA i successes in sensing channel idle for time interval t. (Also, the $P^i_{idle}(t)$ 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 $P^i_{idle}(DIFS)$ and $1 - P^i_{idle}(DIFS)$ respectively. When STA i senses the channel idle in SIFS but not exceeding the DIFS interval time, the STA will still stay in the *Packet_Arrival* state. If STA i does not sense busy in SIFS period but sensing busy in DIFS period, it will transit into *Backoff* state. Before a STA transits from *Packet_Arrival* state to *Attempt* state, it will take DIFS time interval to make sure the channel is idle. Furthermore, the transition from *Packet_Arrival* to *Backoff* state will first at spend maximal DIFS time interval to detect the medium is available or not. In the case of channel busy, it will defer extra *backoff* time $\tilde{b}$.

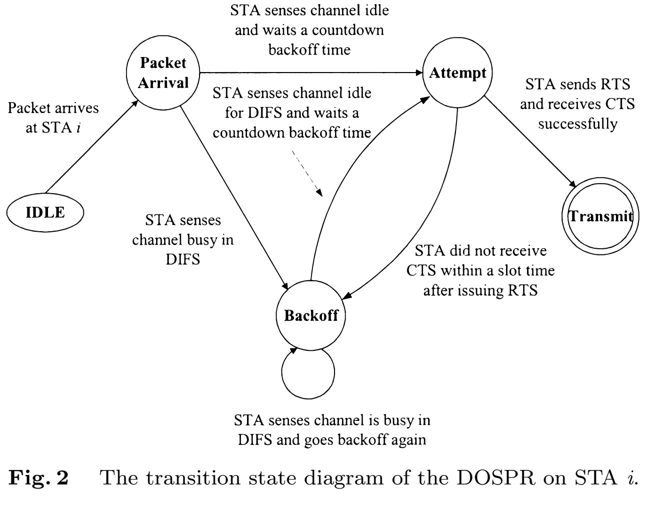

Fig. 2 The transition state diagram of the DOSPR on STA i.

In the *Backoff* state, STA i has the probability $1 - P^i_{idle}(DIFS)$ 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/CCK 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 $P^i_{idle}(DIFS)$), 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 $1-P^i_{idle}(slot)$ and the waste time is RTS + 2×SIFS. On the contrary, STA i has the probability $P^i_{idle}(slot)$ 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 $P^i_{idle}(t)$ 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

$$P_n(t) = \frac{(\lambda t)^n}{n!} e^{-\lambda t}. \tag{1}$$

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

$$P_0(t) = e^{-\lambda t}. \tag{2}$$

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:

$$|Adj(i)| = \sum_{v=1}^{N} s(i,v). \tag{3}$$

From the point of view of STA i, the total packet arrival rate of other stations in cluster is $\hat{\lambda} = |Adj(i)| \times$

λ. Therefore, the probability can be derived as follows:

$$P^i_{idle}(t) = e^{-\hat{\lambda} t}. \tag{4}$$

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

$$\begin{aligned} EA(i) &= P^i_{idle}(slot) \times (RTS + 2 \cdot LIFS + CTS) \\ &\quad + (1 - P^i_{idle}(slot)) \\ &\quad \times (RTS + 2 \cdot SIFS + EB(i)) \end{aligned} \tag{5}$$

and

$$\begin{aligned} EB(i) &= P^i_{idle}(DIFS) \times (DXFS + \tilde{b} + EA(i) \\ &\quad + (1 - P^i_{idle}(DIFS)) \times (\bar{B} + EB(i)). \end{aligned} \tag{6}$$

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

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

$$\begin{aligned} EB(i) &= \frac{1}{P^i_{idle}(DIFS)} \times [P^i_{idle}(DIFS) \\ &\quad \times (DIFS + \tilde{b} + RTS + 2 \cdot SIFS \\ &\quad + P^i_{idle}(slot) \times CTS) \\ &\quad + (1 - P^i_{idle}(DIFS)) \times \bar{B}]. \end{aligned} \tag{7}$$

Now we will solve the parameter $\tilde{b}$ in equation $EB(i)$. Recall symbol $\tilde{b}$ is the mean *backoff* 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 ($0 \le n \le 5$) can be calculated by the following recursive function:

$$\begin{aligned} b(0) &= P^i_{idle}(slot) \times \frac{2^0 \cdot W}{2} + (1 - P^i_{idle}) \times b(1) \\ b(1) &= P^i_{idle}(slot) \times \frac{2^1 \cdot W}{2} + (1 - P^i_{idle}) \times b(2) \\ b(2) &= P^i_{idle}(slot) \times \frac{2^2 \cdot W}{2} + (1 - P^i_{idle}) \times b(3) \\ b(3) &= P^i_{idle}(slot) \times \frac{2^3 \cdot W}{2} + (1 - P^i_{idle}) \times b(4) \\ b(4) &= P^i_{idle}(slot) \times \frac{2^4 \cdot W}{2} + (1 - P^i_{idle}) \times b(5) \\ b(5) &= \frac{2^5 \cdot W}{2} = 2^4 \times W \end{aligned} \tag{8}$$

Then, we obtain

$$\begin{aligned} \tilde{b} &= \sum_{n=0}^{4} (P^i_{idle}(slot) \times (1 - P^i_{idle}(slot))^n \times 2^{n-1} \\ &\quad \times W) + (1 - P^i_{idle}(slot))^5 \times 2^4 \times W. \end{aligned} \tag{9}$$

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

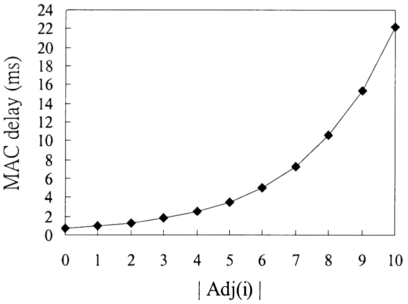

Fig. 3 The expected MAC delay of STA i under different number of neighbor nodes.

$$\begin{aligned} D^i_{idle} &= P^i_{idle}(DIFS) \times (DIFS + \tilde{b} + EA(i)) \\ &\quad + (1 - P^i_{idle}(DIFS)) \times (SIFS + EB(i)) \\ &\quad + packet\_len. \end{aligned} \tag{10}$$

Figure 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 2 Mbps). 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. This is because that a smaller MAC delay implies that 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^i_{delay}. (That is, $s(i,j) = s(i,j) \times D^i_{delay}, \forall 1 \le j \le N$.) The shortest path of the minimal delay can be found by employing the Dijkstra algorithm [5]. (We also note that each element of zero indicates infinite delay cost in Dijkstra algorithm). Let's consider the network shown in Fig. 1 again and use the same assumption with $\lambda = 0.1$ 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

Table 1 System parameters in simulations.

Parameter	Normal Value
Channel bit rate	2 Mbps
Transmission range	200 m
RTS frame length	160 bits
CTS frame length	112 bits
ACK frame length	112 bits
Slot Time (slot)	20 μs
SIFS	10 μs
DIFS	50 μs
PHY and MAC header	400 bits
CWmin	31 slots
CWmax	1023 slots
Propagation delay	1 μs

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.

3. 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 1.

3.1 Simulation Environment

In our simulations, we simulated a scenario of 20 hosts simultaneously active in a square area of 600 m × 600 m. The initial location of each host is assigned randomly. Each host has a transmission range of 200 m. Since the exact route calculation delay of the hybrid approach is hard to determine, we just simplify our simulations without considering the delay for collecting whole network information. 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 moves individually with a move probability. The move probability in simulation is considered from 0.1 to 1.0 in a step of 0.1. The distance of each moving is 100 m 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. In model II, our simulations do not consider the rerouting process. Thus, when the link between two adjacent stations is no longer existing, any packet needs pass through this link will be discarded in our simulation. This means that only one packet needs to pass through a non-existing link from a source will be discarded and all succeeding packets that need to pass through the non-existing link from the source will be rerouted and retransmitted from source by higher layer protocol.

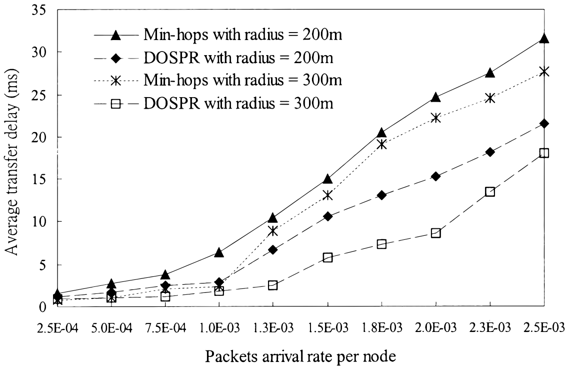

Fig. 4 Comparisons of the derived average transfer delays by DOSPR and Min-hops approach under different packet arrival rate in model I.

Each simulation run is last 200 seconds (≈ 107 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 [6], which is about 50 Bytes–150 Bytes (i.e., about 10 slots–30 slots in 2 Mbps 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 three parameters: the average path length (in hop-count), the average transfer delay and the packet loss rate. The average transfer delay is defined as the average delay, which including the MAC delay, buffer queuing delay and transmission delay, of a packet travelling 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.

3.2 Simulation Results

Figures 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 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 de-

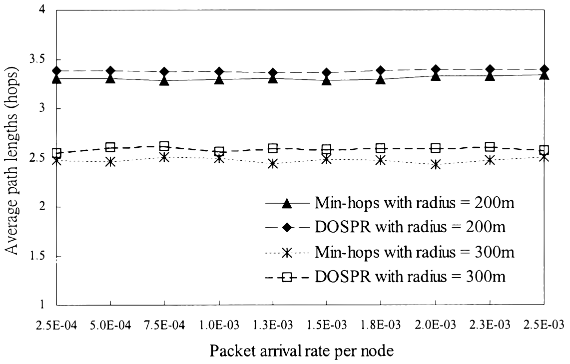

Fig. 5 Comparisons of the derived average path lengths by DOSPR and Min-hops approach under different packet arrival rate in model I.

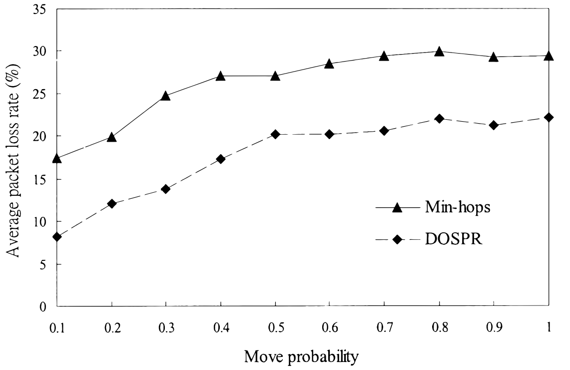

Fig. 6 Comparisons of derived average packet loss rate by DOSPR and Min-hops approach under different packet arrival rates in model II when $\lambda = 10^{-3}$ and $L = 20$ slots.

lay along the selected path. As enlarging the transmission radius to 300 m (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.

Figures 6 and 7 show the DOSPR protocol has the better ability to handle the mobility. Obviously, the derived packet loss ratio by DOSPR is always smaller than that of Min-hops approach in simulation model II. This is because that the DOSPR, in Fig. 6, will select a longer path to obtain less transfer delay; hence, the distance between two adjacent hosts, which was chosen by DOSPR approach, 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. Furthermore, Fig. 7 shows under different move probabilities (50% and 100% respectively) the DOSPR approach will always get the best average packet loss rate than conventional Min-hops approach. These results show the DOSPR protocol also performs very well in the scenario of rapidly changing network topology. Consequently, the Min-hops approach will easily suffer from path loss and need extra rerouting overhead. This is another drawback of Min-hops approach.

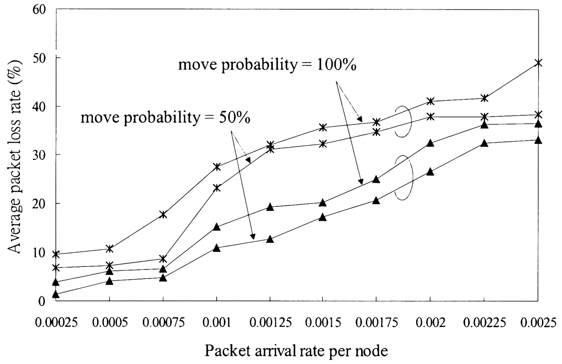

Fig. 7 Comparisons of derived average packet loss rate by DOSPR and Min-hops approach under different packet arrival rates and move probabilities in mode II with transmission radius $= 200$ m and $L = 20$ slots.

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.

4. 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 by proposed DOSPR 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 reduced. 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," Proc. 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," in Mobile Computing, ed. T. Imielinski and H. Korth, chapter 5, pp.153–181, Kluwer Publishing Company, 1996.

[3] C.E. Perkins and E.M. Royer, "Ad-hoc on-demand distance vector routing," Proc. IEEE WMCSA'99, pp.90–100, New Orleans, LA, Feb. 1999.

[4] M.J. Atallah, Algorithms and theory of computation handbook, contents 6-13, CRC Press, 1999.

[5] LAN MAN Standards Committee of the IEEE Computer Society, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 802.11b, ANSI/IEEE Std. edition, Sept. 1999.

[6] K.M. Khalil, K.Q. Luc, and D.V. Wilson, "LAN traffic analysis and workload characterization," Proc. 15th Conference on Local Computer Networks, pp.112–122, 1990.

Shiann-Tsong Sheu was born in Taiwan on September 9, 1968. He received the B.S. degree in applied mathematics from National Chung Hsing University, Taiwan, R.O.C., in 1990, and the Ph.D. degree in computer science from National Tsing Hua University, Taiwan, R.O.C., in 1995. Since 1995 he has been an Associate Professor in the Department of Electrical Engineering, Tamkang University, Taiwan, R.O.C. His current research interests include ATM networks, WDM networks, personal communication networks and design and analysis of protocols and algorithms.

Tihjia Tsai received the BS degree in mechanical from National Taiwan University in 1985 and the MS and Ph.D. degrees in computer science from Michigan State University, East Lansing, in 1993 and 1997 respectively. Currently, he is an Assistant Professor in the Department of Computer Science and Information Engineering, Tamkang University, Taiwan, R.O.C. He is also a coauthor of the CAPA (a Computer Assisted Personalized Approach, for assignments, quizzes, and examinations) system, which is currently in use at Michigan State University and several other universities. His current research interests include communication issues in parallel processing, modelling and analysis of parallel algorithms, distributed computing, network device drivers, and network interfaces for high-speed networks.

Jenhui Chen was born in Taipei, Taiwan on October 12, 1971. He received the B.S. degree from the Department of Computer Science and Information Engineering, Tamkang University, Tamsui, Taipei, Taiwan, R.O.C., in 1998. Currently, he is a candidate of the Ph.D. degree in the Department of Computer Science and Information Engineering at the same university. His current research interests include design, analysis and implementation of network protocols, routing algorithms, wireless communications and WDM networks. He is also a member of IEEE and ACM.