This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the ICC 2008 proceedings.

Finding the Fastest Path in Wireless Networks

Yi Xu and Wenye Wang
Department of Electrical and Computer Engineering
North Carolina State University
Raleigh, NC 27606
Email: {yxu2, wwang}@ncsu.edu

***Abstract*—The timeliness of packet delivery is an important performance measure in wireless networks, especially when urgent messages need to be transported through a network. This paper investigates the fastest packet transportation in light-loaded wireless networks. We show that the end-to-end packet delay depends largely on the locations of the relay nodes that forward the packet and there exists a shortest-delay path theoretically. We also propose a routing algorithm to locate a fast relay path in actual networks to achieve the near-shortest packet delay.**

I. Introduction

Multihop routing is an important strategy for information transmission in wireless ad hoc networks. The nodes in the network cooperate to relay a packet if the source and destination are not within each other's transmission range. In order to establish the multihop path, a routing protocol is needed that may consider a variety of factors, such as shortening the path discovery delay [1], minimizing the routing overhead [2], improving the protocol scalability [3], and saving the node energy [4]. However, the shortest-delay routing is not studied sufficiently in the wireless networks.

We study in this paper the routing strategy to expedite packet transportation in wireless networks by finding the shortest-delay path. Locating the shortest-delay path in a wireless network not only provides the support for express communications, but also enhances our understanding on the best achievable network performance in terms of the packet delay. Previous study on the shortest-delay path discovery is mainly from the load balancing perspective [5], [6], where the path traversing the least loaded nodes is selected such that a packet experiences the least queuing and processing delays at the intermediate relay nodes. In some network scenarios, however, the traffic load is low everywhere, for example the sensor networks. As the task of a sensor network is environmental monitoring, there are few message exchanges most of the time when no concerned event occurs in the area of surveillance. When events happen and reports are sent to the sink, the load on the relay nodes is also well below the processing and the communication capability of these nodes, because the reports are generated only from the limited number of sensors near the event locations and the sensors are designed to handle such amount of traffic without difficulty. In these light-loaded networks, load balancing does not gain noticeable improvement on the packet transportation delay. Instead, the wireless link capacity becomes the dominant factor in determining the packet delay.

This work was supported by the NSF CAREER Award under Grant Number CNS-0546289.

The well-known *Shannon Capacity* defines the maximum bandwidth that a communication link can provide, which is expressed as $C = B\log_2(1+\frac{S}{N})$, where C is the available bandwidth, B is the frequency band, and $\frac{S}{N}$ is the signal-to-noise ratio at the receiver. For wireless links, due to path loss, the received signal at a receiver is $S = Pd^{-\alpha}$, in which P is the transmitted power, d is the Euclidean distance between the sender and the receiver, and α is the attenuation exponent. Thus, the length d of a wireless link directly affects the link capacity, which in turn determines the transmission delay of a packet on this link. The end-to-end delay experienced by a packet, which is the summation of all the transmission delays on the traversed links, thus depends on the specific path chosen by the packet. As such, identifying the relay path that has the shortest transmission delay is the key to achieving the fastest packet transportation in light-loaded wireless networks.

In order to guide the path search, we define the concept of *Packet Transportation Speed*. Minimizing the end-to-end packet transmission delay is equivalent to maximizing this packet transportation speed. Investigation shows that this speed has an upper bound that is reached under certain conditions. We identify these conditions and apply them as the guidance for path search through designing a new routing algorithm. This routing algorithm finds the near-optimal relay path in terms of the shortest end-to-end packet delay.

The rest of this paper is organized as follows. We formulate the packet transportation speed problem and identify the conditions for the maximum speed in Section II. Based on the theoretical result, we propose in Section III a routing algorithm that locates the shortest-delay path in light-loaded random wireless networks, the performance of which is evaluated in Section IV. Finally, Section V concludes this paper.

II. Fastest Packet Transmission

A. Problem Formulation and Assumptions

We define the *Packet Transportation Speed* as follows to measure how fast a packet is transmitted in a network:

$$v(t) = \frac{d(t)}{t}, \tag{1}$$

where t is the time duration since the packet has left its source node and $d(t)$ is the straightline distance travelled by the packet during the period t. In this paper, we study

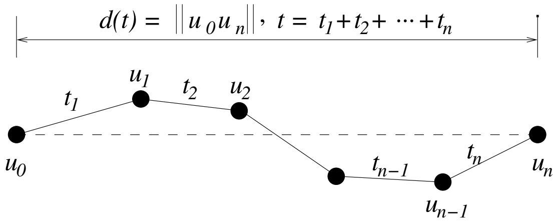


Fig. 1. An example of multihop transmission, where a packet is transmitted from u_0 to u_n in n hops with transmission delay t_i on the i-th hop. $\|u_0 u_n\|$ denotes the straightline distance between u_0 and u_n.

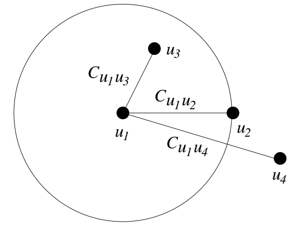


Fig. 2. An illustration of the transmission radius x^*, where $\|u_1u_2\| = x^*$, $\|u_1u_3\| < x^*$, and $\|u_1u_4\| > x^*$. As $C_{u_iu_j} = B\log_2(1+\frac{P}{N}\|u_iu_j\|^{-\alpha})$, $C_{u_1u_3} > C_{u_1u_2} > C_{u_1u_4}$. When u_1 sends a packet to u_2 in the rate $C_{u_1u_2}$, u_3 receives the same packet correctly, but u_4 does not.

a light-loaded network where the packet queuing, processing and collision-incurred retransmission delays are negligible as compared to the packet transmission delay. We also ignore the propagation delay of the electrical wave that travels in the speed of light in the air. As such, the duration t may be written as $t = \sum_{i=1}^{n} t_i$, where t_i is the transmission delay over the i-th hop. An example of the multihop transmission is illustrated in Fig. 1. For a given pair of source and destination nodes u_0 and u_d, $d(t) = \|u_0 u_d\|$ is fixed. Hence, minimizing the end-to-end transmission delay t is equivalent to maximizing the speed $v(t)$, which allows us to transform the packet delay minimization problem into the packet transportation speed maximization problem. The solution for the latter problem will simultaneously serve the former problem.

Next, we show that there exists an upper bound v^* such that $v(t) \le v^*$ for any relay path. For clearer presentation, we first list our assumptions and notations below.

- The nodes are static and randomly located in the network in a Poisson point process with density λ.
- A total of B Hz spectrum is shared by all the nodes.
- Every node has a fixed transmission power P.
- The noise N including the ambient and the interference noise is constant everywhere in the network.
- The wireless link between two nodes is characterized by a path loss model with the attenuation exponent $\alpha \ge 2$ [7]. Some typical values of α include $\alpha = 2$ as in the free space and $\alpha = 4$ as in the urban area.
- The capacity of a wireless link is $C = B\log_2(1+\frac{P}{N}d^{-\alpha})$, where d is the Euclidean length of the link.
- Advanced error control codes are used such that the available link bandwidth equates its capacity C.
- A packet contains L bits.

B. Fastest Packet Transportation Speed

Let us denote $\|u_{i-1}u_i\|$ as the Euclidean distance between nodes u_{i-1} and u_i, and t_i as the time to transmit a packet from u_{i-1} directly to u_i. By definition,

$$
\begin{aligned}
v(t) &= \frac{d(t)}{t} = \frac{\|u_0u_n\|}{t} \\
&\le \frac{\sum_{i=1}^{n}\|u_{i-1}u_i\|}{\sum_{i=1}^{n} t_i} &(2)\\
&\le \max_i \frac{\|u_{i-1}u_i\|}{t_i} &(3)\\
&= \max_i \frac{B}{L}\|u_{i-1}u_i\|\log_2(1+\frac{P}{N}\|u_{i-1}u_i\|^{-\alpha}) \\
&\le \frac{B}{L}x^*\log_2(1+\frac{P}{N}(x^*)^{-\alpha}). &(4)
\end{aligned}
$$

It is shown in [8] that there exists a maximizer x^* to the function $x\log_2(1+\frac{P}{N}x^{-\alpha})$. Hence, $v(t) \le v^*$, where $v^* = \frac{B}{L}x^*\log_2(1+\frac{P}{N}(x^*)^{-\alpha})$. In order to achieve v^*, all the Inequalities (2), (3) and (4) must be tight, which takes place when all of the following conditions are satisfied:

1) The relay nodes are located in a straightline such that $\|u_0u_n\| = \sum_{i=1}^{n}\|u_{i-1}u_i\|$ (see Inequality (2)).
2) The neighboring relay nodes are separated from each other in equal distance such that $\frac{\|u_0u_1\|}{t_1} = \cdots = \frac{\|u_{n-1}u_n\|}{t_n}$ (see Inequality (3)).
3) The separation distance equals x^*, i.e., $\|u_{i-1}u_i\| = x^*$ $(i = 1, \cdots, n)$ (see Inequality (4)).

These optimal conditions identify a packet relay path that transmits a packet at the fastest speed.

C. Transmission Radius (x^*) Determination

Note that when a node forwards a packet to the next node located in x^* distance away using the full capacity of the wireless link between them, any node located in a distance less than x^* from the sender can also receive the packet correctly, but any node located more than x^* distance away from the sender cannot receive the packet. Fig. 2 depicts an example of such situation. As such, we name x^* the *transmission radius* in the sense of achieving the fastest packet transmission.

The value of x^* is critical, as it determines the desired location of the next-hop relay node as well as the data rate a forwarding node can use. However, its solution is not given in [8]. We derive its solution next. By defining $f(x) = x\log_2(1+\frac{P}{N}x^{-\alpha})$, we know that x^* satisfies the equation $f'(x^*) = 0$, which is expressed as

$$(1+y)\log_2(1+y) = \frac{\alpha}{\ln 2}y, \tag{5}$$

where $y = \frac{P}{N}(x^*)^{-\alpha}$. Equation (5) can be rewritten as

$$y = e^{\frac{\alpha y}{1+y}} - 1, \tag{6}$$

which allows us to compute y in a recursive way:

$$
\begin{cases}
y_0 = e^{\alpha} - 1, \\
y_i = e^{\frac{\alpha y_{i-1}}{1+y_{i-1}}} - 1 \qquad i = 1, 2, \cdots .
\end{cases}
\tag{7}
$$

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the ICC 2008 proceedings.

TABLE I
SAMPLE VALUES OF $y(\alpha)$ AND $k(\alpha)$

α	$y(\alpha)$	$e^\alpha - 1$	$k(\alpha)$	$(\frac{1}{e^\alpha-1})^{\frac{1}{\alpha}}$
2	3.9216	6.3891	0.5050	0.3956
3	15.8010	19.0855	0.3985	0.3742
4	49.4353	53.5982	0.3771	0.3696
5	142.3249	147.4132	0.3710	0.3684
6	396.3833	402.4288	0.3690	0.3680

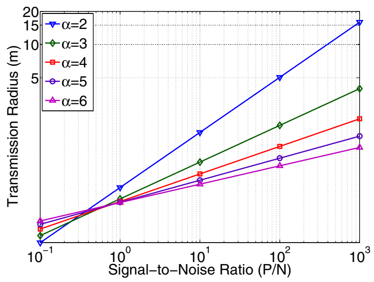


Fig. 3. The values of transmission radius x^*.

The final value of y_i after it has converged is the solution to Equation (5). Since y depends on α, we write $y = y(\alpha)$. By defining $k(\alpha) = (\frac{1}{y(\alpha)})^{\frac{1}{\alpha}}$, we obtain

$$x^* = (\frac{P}{Ny(\alpha)})^{\frac{1}{\alpha}} = k(\alpha)(\frac{P}{N})^{\frac{1}{\alpha}}. \quad (8)$$

Table I lists the computed $y(\alpha)$ and $k(\alpha)$ for some sample α values, in which we also compute $e^\alpha - 1$ and $(\frac{1}{e^\alpha-1})^{\frac{1}{\alpha}}$ as the closed-form approximation to $y(\alpha)$ and $k(\alpha)$. We see that $y(\alpha)$ and $k(\alpha)$ can be approximated well by $e^\alpha - 1$ and $(\frac{1}{e^\alpha-1})^{\frac{1}{\alpha}}$, especially for large values of α. Thus, for any given α, P and N, we can either determine $y(\alpha)$, $k(\alpha)$ and x^* by Equations (7) and (8) or approximate x^* as

$$x^* \approx (\frac{P}{N(e^\alpha - 1)})^{\frac{1}{\alpha}}. \quad (9)$$

Some sample values of x^* are plotted in Fig. 3 with respect to different signal-to-noise ratios and different α values. Next, we apply the result of x^* to design a new routing algorithm, the goal of which is to discover the path with the shortest end-to-end transmission delay.

III. THE FASTEST PACKET TRANSMISSION ALGORITHM

Based on the analysis, we know that a packet is transmitted at the fastest speed if the next-hop node is located at a distance of x^* from the forwarding node and in the direction towards the destination. In this section, we design a routing algorithm that identifies the next-hop relay nodes to achieve the fastest packet transmission. As this new algorithm assumes the knowledge of node locations, it can be viewed as a variant of the geographic routing algorithms. Specifically, we assume the following information is available to every node [9]:

- the location of the node itself,
- the location of the packet destination,

The Fastest Packet Transmission Algorithm

find x^* by Equations (7) and (8) or by Equation (9)
determine $\gamma = \frac{x^*}{\|u_i u_d\|}$
if $\gamma \geq 1$
  then compute $C = B\log_2(1 + \frac{P}{N}\|u_i u_d\|^{-\alpha})$
    transmit the packet to u_d at rate C
  else determine $\mathbf{z} = (1-\gamma)\mathbf{u_i} + \gamma\mathbf{u_d}$
    find node $u_{i+1} = \arg_u \min\{\|zu\| : \|zu\| < \delta\}$
    compute $C = B\log_2(1 + \frac{P}{N}\|u_i u_{i+1}\|^{-\alpha})$
    transmit the packet to u_{i+1} at rate C

Fig. 4. The fastest packet transmission algorithm running on node u_i that forwards a packet to destination u_d. The vectors $\mathbf{u_i}$ and $\mathbf{u_d}$ denote the locations of u_i and u_d. The environmental parameters α, B, P and N are constant and known. δ defines the radius of the region in which the next-hop relay node is searched for.

- the locations of the neighbor nodes that are defined by a neighborhood radius ρ.

We comment briefly on the availability of node locations. In static networks, since the nodes do not move, each node can be configured to be aware of its own location. The locations of packet destinations are known in some scenarios, e.g. in sensor networks, where the designated data sinks that collect event reports from the other nodes have publicly known locations. Furthermore, a node is able to learn the locations of its neighbors through local information exchange.

By assuming the availability of location information, we design a fastest packet transmission algorithm as in Fig. 4, which runs on every node in the network distributively and independently. To identify the next-hop node, a forwarding node u_i finds out the transmission radius x^* first. x^* can be either accurately determined by using the Equations (7) and (8) or approximated by the Equation (9). Next, if $x^* \geq \|u_i u_d\|$ where u_d is the destination node, the packet is sent directly to u_d. In this case, u_i computes the capacity C of the wireless link connecting itself and the destination node u_d, and then sends the packet at the computed bit rate C. Higher rate than C will cause incorrect reception at u_d and lower rate than C will introduce extra unnessary transmission delay. Otherwise, if $x^* < \|u_i u_d\|$, at least one more relay node is needed. In this case, u_i first determines the desired location of the next-hop node, then finds the node closest to the desired location, and finally computes the link capacity to send the packet at the correct rate to the identified next-hop node.

Note that the neighborhood radius ρ and the search region radius δ need to be determined for the algorithm completeness. In order to minimize the scope of local information exchange (which incurs overhead) and to guarantee that a packet is transmitted in the correct direction, ρ and δ are determined as follows. To ensure the packet is approaching the destination u_d in each hop, the following must hold:

$$\|u_{i+1}u_d\| < \|u_i u_d\|. \quad (10)$$

From Fig. 5, the triangle inequality requires

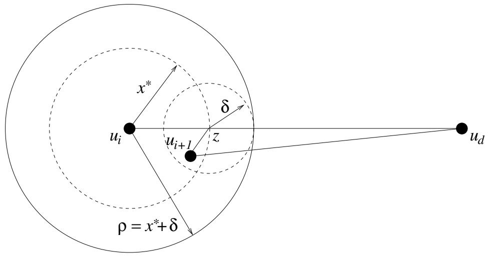


Fig. 5. The neighborhood radius ρ and the search region radius δ.

$$\begin{aligned} \|u_{i+1}u_d\| &\leq \|zu_d\| + \|zu_{i+1}\| \\ &< \|zu_d\| + \delta. \end{aligned} \tag{11}$$

Therefore, if $\|zu_d\| + \delta \leq \|u_i u_d\|$, which is equivalent to

$$\delta \leq \|zu_i\| = x^*, \tag{12}$$

the packet is guaranteed to be forwarded in the correct direction. Because large δ reduces the likelihood of finding no relay node in the search region, we choose $\delta = x^*$. Considering the arbitrary location of u_d, u_i needs to know the locations of all the nodes within a neighborhood radius $\rho = x^* + \delta = 2x^*$.

However, there is a special case that we need to discuss on the proposed algorithm. If the network is sparse, there may not exist any node in the search region defined by the radius δ. Should this case occur, u_i may enlarge its search region until it finds a node that is closer to u_d than itself. In the extreme case that u_i still cannot find any next-hop node before u_d is covered by its enlarged search region, u_i sends the packet directly to u_d.

IV. Algorithm Evaluation

We evaluate the performance of our proposed algorithm in this section. As it assumes the availability of node location information, we first compare its performance with the other geographic routing algorithms which also use node location information.

A. Comparison to Geographic Routing Algorithms

All the geographic routing algorithms select the next-hop relay node based on the relative locations of the forwarding node, its neighbors and the packet destination. We will consider in particular the following two algorithms in this paper.

- *Greedy Routing.* The greedy routing algorithm chooses the neighbor node closest to the destination to be the next-hop relay node [9], i.e., $u_{i+1} = \arg_u \min\{\|uu_d\| : \|uu_i\| < r\}$, where r defines the radius of the neighborhood. There also exist other algorithms similar to the greedy routing, in which the packet forwarding progress is measured by different metrics and the node resulting in the largest one-hop progress is selected as the next relay node. We use the greedy routing algorithm as proposed in [9] to represent this broader category of algorithms.
- *Random Routing.* We introduce the random routing algorithm to represent a probabilistic version of the greedy algorithms. Rather than deterministic selection, it chooses the next-hop relay node according to a probability distribution. The probability to choose node u is defined to be $\Pr[u] = \frac{w(u)}{\sum_{\{u\}} w(u)}$, where $w(u) = \|u_i u_d\| - \|uu_d\|$ and $\{u\} = \{u : w(u) > 0, \|uu_i\| < r\}$.

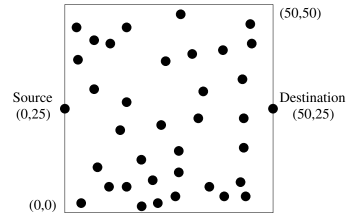


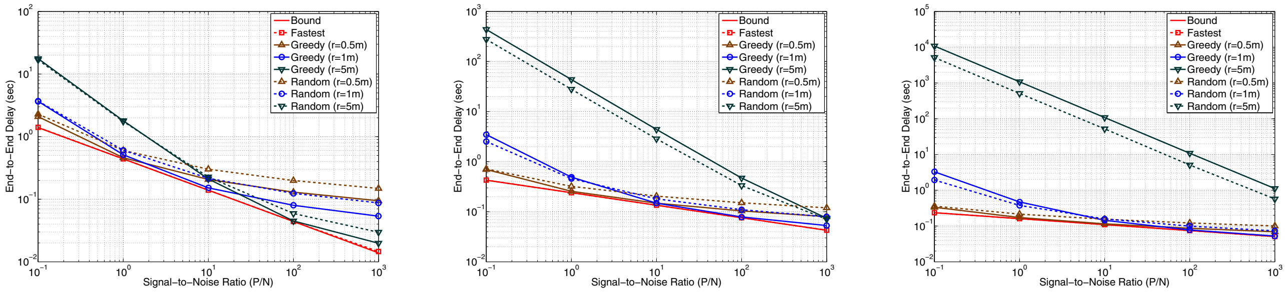
Fig. 6. The simulation network.

We simulate a network of area 50×50 m^2, as shown in Fig. 6, in which the nodes are randomly distributed in a Poisson point process with density $\lambda = 100$ $\mathrm{nodes/m}^2$. All the nodes share a total of $B = 100$KHz frequency bandwidth. The source and the destination nodes are located at the coordinates $(0, 25)$ and $(50, 25)$ respectively. In the simulation, we let the source node send a packet of length $L = 128$ bytes to the destination node by using our proposed transmission algorithm as well as the greedy and the random geographic routing algorithms. We vary the values of α, $\frac{P}{N}$ and r, and in each setting an algorithm is simulated for 500 times to obtain the average end-to-end transmission delay of the packet. The results of the packet delay are plotted in Fig. 7. As a reference, we have also included the theoretical bound of the shortest transmission delay when the packet is transmitted along a perfect path in which all the relay nodes are located at the desired locations.

We observe that the theoretical delay bound decreases as the signal-to-noise ratio increases, because the improvement on the wireless channel condition allows higher link capacity and faster packet transmission. Our proposed fastest transmission algorithm achieves almost the same performance as the theoretical delay bound in all the simulation settings. This is because we can always find a relay node very close to the desired location when the node density is high (100 $\mathrm{nodes/m}^2$ in the simulation). On the contrary, the greedy and the random algorithms result in obviously longer packet delay. We notice that, however, the greedy and the random algorithms approach the delay bound closely at some points in the figures when they happen to choose a neighborhood radius close to the value of x^*. In such cases, given high node density, all these algorithms find similar relay paths.

B. The Delay Gap

Since it is impossible to find the perfectly located relay nodes due to location randomness, there is always a gap between the actual packet delay and the delay bound. Intuitively, higher node density reduces the gap by allowing us to find better relay nodes that are closer to the desired locations. As shown in Fig. 7 where node density is high, the gap is not noticeable. To be more accurate, the delay gap satisfies the



(a) $\alpha = 2$ (b) $\alpha = 4$ (c) $\alpha = 6$

Fig. 7. The end-to-end packet delay.

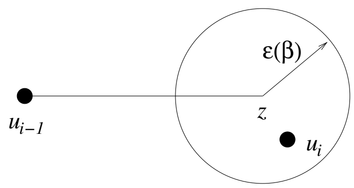


Fig. 8. The i-th hop transmission, where $\|zu_{i-1}\| = x^*$ and $\epsilon(\beta)$ is given by Equation (14).

following inequality:

$$\Pr[t \le \beta t^*] \ge \left(1 - e^{-\lambda\pi\epsilon^2(\beta)}\right)^{n-1}, \tag{13}$$

where

$$\epsilon(\beta) = \left(\frac{P}{N((1+y(\alpha))^{\frac{1}{\beta}} - 1)}\right)^{\frac{1}{\alpha}} - \left(\frac{P}{Ny(\alpha)}\right)^{\frac{1}{\alpha}}, \tag{14}$$

t is the actual transmission delay through an n-hop relay path, t^* is the delay bound achieved when all the $n-1$ relay nodes are perfectly located, and $\beta > 1$ is a factor to measure the gap between t and t^*.

To prove (13), we first show that

$$\Pr[t_i \le \beta t_i^*] \ge 1 - e^{-\lambda\pi\epsilon^2(\beta)}, \tag{15}$$

where t_i is the actual transmission delay on the i-th hop (from u_{i-1} to u_i as in Fig. 8) and $t_i^* = \frac{L}{B\log_2(1+\frac{P}{N}(x^*)^{-\alpha})}$ is the ideal delay if node u_i is perfectly located (at point z as in Fig. 8). From Fig. 8, if $\|zu_i\| \le \epsilon(\beta)$, $\|u_{i-1}u_i\| \le x^* + \epsilon(\beta)$ and $t_i = \frac{L}{B\log_2(1+\frac{P}{N}\|u_{i-1}u_i\|^{-\alpha})} \le \frac{L}{B\log_2(1+\frac{P}{N}(x^*+\epsilon(\beta))^{-\alpha})}$. As Equation (14) can be equivalently changed into

$$\frac{L}{B\log_2(1+\frac{P}{N}(x^*+\epsilon(\beta))^{-\alpha})} = \beta \cdot \frac{L}{B\log_2(1+\frac{P}{N}(x^*)^{-\alpha})}, \tag{16}$$

we obtain $t_i \le \beta t_i^*$. This result demonstrates that as long as we can find a node within distance $\epsilon(\beta)$ from point z, which occurs with probability $1 - e^{-\lambda\pi\epsilon^2(\beta)}$ according to the Poisson point process, $t_i \le \beta t_i^*$ holds. Considering the fact that finding such a node is a sufficient but not necessary condition for $t_i \le \beta t_i^*$, $\Pr[t_i \le \beta t_i^*] \ge 1 - e^{-\lambda\pi\epsilon^2(\beta)}$. Now consider an n-hop path. If all the $n-1$ relay nodes are located within distance $\epsilon(\beta)$ from their respective desired locations, $t = \sum_{i=1}^n t_i \le n\beta t_i^* = \beta t^*$. Again, since this is a sufficient but not necessary condition, we have $\Pr[t \le \beta t^*] \ge (1 - e^{-\lambda\pi\epsilon^2(\beta)})^{n-1}$. This result indicates that the gap between t and t^* is bounded by a constant $(\beta - 1)t^*$ with higher probability when λ grows. Thus on average the gap reduces as node density increases.

V. Conclusions

In this paper we have studied the problem of finding the shortest-delay path in light-loaded wireless networks from the link capacity perspective. Due to the correlation between link capacity and link distance in the wireless environment, the end-to-end packet transmission delay is determined by the locations of the intermediate relay nodes. In order to identify the fastest path, we have first determined the desired locations of the relay nodes from a mathematical model. Then we have designed a routing algorithm to discover this fastest path. Investigation demonstrates that this new routing algorithm can successfully transport a packet in the near-shortest time, especially when node density is high.

References

[1] C.E. Perkins, P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance Vector (DSDV) for Mobile Computers", *Proc. of ACM SIGCOMM*, pages 234–244, August 1994.

[2] C. Perkins, E. Belding, S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing", IETF RFC 3561, http://www.ietf.org.

[3] G. Xing, C. Lu, R. Pless, Q. Huang, "On Greedy Geographic Routing Algorithms in Sensing-Covered Networks", *Proc. of ACM MobiHoc*, May 2004.

[4] S. Banerjee, A. Misra, "Minimum Energy Paths for Reliable Communication in Multi-hop Wireless Networks", *Proc. of ACM MobiHoc*, June 2002.

[5] H. Hassanein, A. Zhou, "Routing with Load Balancing in Wireless Ad hoc Networks", *Proc. of ACM MSWiM*, July 2001.

[6] Y. Ganjali, A. Keshavarzian, "Load Balancing in Ad Hoc Networks: Single-Path Routing vs. Multi-Path Routing", *Proc. of IEEE INFOCOM*, March 2004.

[7] M. Schwartz, *Mobile Wireless Communications*, Cambridge University Press, 2005.

[8] R. Zheng, "Information Dissemination in Power-Constrained Wireless Networks", *Proc. of IEEE INFOCOM*, April 2006.

[9] B. Karp, H.T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks", *Proc. of ACM MobiCom*, August 2000.