Travel delay in a large wireless ad hoc network

Erol Gelenbe
Dennis Gabor Chair
Dept. of Electrical & Electronic Engineering
Imperial College
London SW7 2BT
e.gelenbe@imperial.ac.uk

***Abstract*— An analysis based on Brownian Motion, including the effect of packet losses, packet retransmission and the overhead delay incurred after each time-out, is used to derive closed form expressions for the average travel time of a packet from its source to a fixed destination. An iterative algorithm is proposed to compute the time it takes for a packet to reach a moving destination node. The value of the optimum time-out interval that minimises average travel time is illustrated with numerical examples.**

I. INTRODUCTION

Many wireless environments cannot rely on a "centralised transmitter-receiver" scheme due to cost considerations (as in cheap and ubiquitous sensor networks), or because such systems need to be deployed in environments where a fixed backbone infrastructure, such as the one available to conventional mobile telephony, cannot be installed or depended upon. Examples include emergency situations where a network has to be deployed when the infrastructure has been destroyed, military networks with mobile nodes, or very low power wireless sensor networks that are fielded in a variety of indoor or outdoor contexts to sense ambient conditions such as temperature, humidity, chemical fumes, noise, and so on.

Such networks have to operate reliably despite the fact that network nodes themselves may be unreliable due to electronic failures, limitations on available battery power or ambient conditions on wireless propagation such as humdity and antenna position or orientation, when the nodes themselves are mobile, and the wireless communication medium is noisy or partially obstructed by physical objects. Thus wireless ad hoc networks and sensor networks typically rely on multiple nodes operating in multi-hop mode to convey packets or other network traffic.

These complex and novel network applications have motivated much interesting work on techniques for forwarding packets which are designed to mitigate for the intrinsic unreliability of multi-hop wireless environments [7], [8], [9], [10].

In this paper we consider a packet wireless network in which nodes are distributed at random over some large area. We assume that the wireless range d of any node is typically much smaller than the distance between two randomly located nodes in the area, so that packet transmission is based on a store-and-forward multi-hop protocol. We also assume that the routing algorithms use directional or geographical information about the destination, as in [5], [8], [11] in order to forward packets, including the effect of imprecision in this information due to a variety of reasons. The problem we study is also of practical interest in wired networks which use local search techniques for packet routing [14] without having the benefit of routing tables for known network paths.

A. Problem setting

Suppose that a node N located (say) at position (x, y) wishes to transmit one or more packets to a node M, and that N initially has knowledge about $M's$ location, call it (x', y'). Both the destination node's identity and it's known location (x', y') (which may be innacurate) are stored in the packet. Then N attempts to transmit the packet, and M receives and acknowledges the packet if it is within wireless hearing distance of N. If on the other hand M is out of direct wireless reach of N, then the transmission proceeds as follows:

- Either there is no node in the wireless vicinity of N, and the packet transmission is stalled. Then N tries again after some time-out period of average value R.
- Or else there is at least one other node within hearing distance of N, and N allocates the packet to that node among this set which is closest to (x', y'). That node will then take on the role of forwarding the packet and it will repeat the process in the same manner.

The question about this scheme that we will try to address in this paper is the following. How can one construct a model of packet propagation in the random medium we have described, that allows us to estimate how long it will take the packet to reach its destination, in the presence of possible mobility of the destination node. An approximate analysis as well as bounds for the number of hops travelled by a packet in an ad hoc network have been previously obtained [12], [13] using a spatial Poisson assumption about the location of the nodes and their number.

In this paper we obtain closed form expressions for the time it takes for a packet to reach a fixed destination node using a Brownian motion (diffusion process) model for the manner in which the distance of the distance changes over time. We also derive a simple condition that must be satisfied if the packet is to reach a moving destination node, and an iterative algorithm for the average time needed to reach the mobile destination.

II. Modelling the Distance of the Packet to its Destination

Consider the distance Y_t of the packet at time $t \geq 0$ from its final destination. We will assume that at time $t = 0$ the distance is $Y_0 = D$. Thus if the destination point is (x, y), and the packet's location at time t is (x', y'), then $Y_t = \sqrt{(x-x')^2 + (y-y')^2}$. We will assume that Y_t is a random variable with average value $E[Y_t] = bt + D$. Thus we do not dwell on the detailed manner in which the mobile node moves, but simply assume its motion has some known average behaviour represented by a net average speed b towards the destination. Note that in general b may be positive, zero or negative, depending on whether the packet or the intermediate nodes have some knowledge of how or in which direction the packet should be forwarded. In this paper, the packet's distance from its destination will be modelled by a Brownian motion which will have both a "drift" parameter b and a second moment parameter $c \geq 0$.

Thus $b\Delta t$ is the mean distance travelled by the packet in time $[t, t+\Delta t)$, while $c\Delta t$ the variance of the distance travelled by the packet in time $[t, t+\Delta)$, where we assume that both b and c do not depend on time, nor on the location of the packet (i.e., they are constants):

$$b = \lim_{\Delta t \to 0} \frac{E[Y_{t+\Delta t} - Y_t | Y_t = y]}{\Delta t},$$

$$c = \lim_{\Delta t \to 0} \frac{E[(Y_{t+\Delta t} - Y_t)^2 | Y_t = y] - (E[Y_{t+\Delta t} - Y_t | Y_t = y])^2}{\Delta t}.$$

Note that $c \geq 0$, with $c = 0$ meaning that the packet has a deterministic motion.

If $b = 0$ this means that the "packet does not know where it is going", i.e., it is drifting without a sense of the good direction. If $b < 0$ then the packet is being mostly sent forward in the right direction towards its destination, while if $b > 0$ then on the average, the packet is moving away from its destination (the most unfavourable case). Furthermore, we also allow for the case where the packet is lost during transmission or not received by a repeater:

- With probability $\lambda\Delta t + o(\Delta t)$ the packet <u>is lost</u>, either because there are no nodes to which the current node can forward the packet, or because of some transmission failure or error. It may also correspond to an absorption or strong attenuation of the signal due to obstacles, or to the fact that the closest repeater is out of range. In that case, the packet will be re-transmitted after a time-out interval of average value $R = 1/r$.
- With probability $r\Delta t + o(\Delta t)$ the sender makes use of its time-out; this can also cover situations in which the packet contains a fixed probability of self-destroying. This probability may be quite small but could be used to abort the packet if it has been travelling through the network for a very long time. Once the packet aborts itself, the transmitting node becomes aware of it after some time and retransmits the packet after a time of average value $m = \mu^{-1}$ which represents the overhead delay.
- The role of r and m merit some further explanation. $r\Delta t$ is the probability with which the sender decides is able to inform the node that is holding the packet that the packet that it has been aborted. At that point the packet enters the "lost" or destroyed state. If the packet was already lost at that time, it is the probability that the sender decides to assume that its packet is indeed lost. However we do not assume that the sender immediately sends out a new copy of the packet; this only happens after a further delay of average value m.
- Finally, with probability $1 - \lambda\Delta - r\Delta t + o(\Delta t)$ the packet <u>is correctly transmitted and received</u> by the next node; thus the packet has now reached a new node whose distance from the destination is $Y_{t+\Delta t}$.

Thus after a successful transmission-reception of the packet, the packet's distance to the destination changes from Y_t to $Y_{t+\Delta t}$. Note however that the packet may now be nearer or further away from the destination. Indeed, if a repeater is not available to take the packet *closer* to the destination, the transmitting node may choose to hand it over to a node that is further away rather than keeping it and re-trying some time later.

One can legitimately ask how the sender may know that a packet is lost. Notice that if λ is quite high (fairly large packet loss rate) and r is very small (very long time-out period), the sender can hear that the packet has been successfully transmitted before it decides to activate its time-out. Indeed, there will be a "back pressure of information" about packets from wherever they may be back towards the source via ACK packets that are sent back at each step, or sent back from the desyination itself. Thus it is not unreasonable to assume that the sender actually can have a pretty good indication about whether his packet is lost, or whether it is making good progress towards the destination (or has actually reached the destination). On the other hand, the question arises about what happens to the packet if it is not destroyed *and* the time-out is invoked and a new packet is sent out. One way of looking at this is to say that if the time-out is big enough (i.e. r is small enough) then this will hardly ever happen. Another way of looking at it is to say that it does not really matter: in this case the average travel time of the packet that we derive is simply an *upper bound* to the real average travel time, since a previous copy of the packet may arrive to the destination *before* the new packet that has been sent out does arrive.

It is of interest to understand the relative magnitudes of the parameters we are dealing with:

- Since the nodes which are within hearing distance of each other may be distant by a few centimeters to a kilometer from each other, if we assume a net packet transmission time of one micro-second for a node, we would expect to have that b is in the range of 10^4 to 10^{10} meters-per-second.
- On the other hand, the destination node may also be

moving, say at speed γ. The order of magnitude may be of one (for human beings moving slowly) to one thousand (for aircraft) kilometers-per-hour, so that γ may be in the range of 0.3 to 300 meters-per-second.

Thus the packet that is being conveyed by the wireless multi-hop medium is "chasing" the destination node. But because of the very high relative speed of packet motion with respect to the motion of mobiles, we can expect that the system can work effectively despite the possibility of having high packet loss rates due to noise or other disturbances in the network which will require multiple attempts at sending out the packet.

A. Will the packet ever reach a mobile destination?

Let us first consider a destination node M which does not move. Since $Y_0 = D$, then the packet reaches the destination at some time T_1 where:

$$T_1 = inf\{t : Y_t = 0\}, \tag{1}$$

In order to determine whether the packet ever reaches its *mobile* destination, we have to consider the manner in which it "chases down" the mobile node that it is trying to reach. The packet first takes time T_1 to reach the initial targeted destination (x', y'). But by the time it reaches it, the mobile will have moved away from (x', y') by an average distance γT_1 from (x', y'). Now the packet pursues the mobile at its new location, and so on. Thus the packet will eventually reach the mobile node M if the average speed with which it moves towards its designated destination is greater than the average speed with which the mobile is able to move away in the same time interval. This condition is simply:

$$\gamma < \frac{D}{E[T_1(D)]}. \tag{2}$$

where we have indicated that T_1 depends on the distance D that the packet travels. Thus in the sequel we first compute $E[T_1(D)]$, which assumes that the destination is fixed.

III. Average packet travel time to a fixed destination

In order to avoid having to compute the properties of the transient process of *"going from the source to the destination just once"*, we will construct an ergodic process where *"the packet goes from the source to the destination, stops there for a short time, and then the travel restarts and is repeated indefinitely"*. This will allow us to calculate $E[T_1(D)]$ more easily, by replacing the computation of the first passage time T_1 in the process Y by a simpler analysis which will yield the expected value $E[T_1]$ from the ergodic process Z which is defined below. Note that the purpose of the ergodic process is to simplify the analysis we have to undertake, and in particular to allow us to compute $E[T_1]$ more easily.

Consider a new random process $Z = \{Z_t : \ t \geq 0\}$ which is identical in sample path to Y until time T_1. After T_1 the process Z resides at the point $z = 0$ (the destination location (x', y')) for a random time H_1, after which Z jumps to point D (the starting location for the packet's travel) and then stochastically repeats its previous behaviour indefinitely. Suppose now that at some time $t = T_i$ the destination is reached, i.e. $Z(T_i) = 0$; then after a random time H_i we assume that the search process starts again, so that at $t = T_i + H_i$ the process Z jumps back to the starting point of the search and $Z((T_i + H_i)^+) = D$, and the search process is re-initialised. This process repeats itself indefinitely.

Let the sequence of H_i, $i = 1, 2, \ ..$, be independent and identically distributed positive random variables, and $T_{i+1} = inf\{t : \ t > T_i, \ Z_t = 0\}$ for $i = 1, 2, \ ...$ Then Z will have the renewal property:

$$P[Z(t) > z] = P[Z(t + T_i + H_i) > z], \tag{3}$$

and for any $t \geq 0$, $z \geq 0$, and the instants $\{T_i + H_i\}$ are renewal instants of the process Z for $i \geq 1$.

Note that the H_i are dummy variables that are mainly chosen to create a pause and repetition; it is simplest to take $E[H_i] = 1$ in order to compute $E[T_1]$ as discussed below.

We represent the "end point" of the packet travel time by a probability mass $P(t)$ at point $z = 0$, so that $P(t)$ is the probability that the packet is at its destination at time t, i.e. $P(t) = \sum_{i=1}^{\infty} P[T_i \leq t < T_i + H_i]$.

If the artificial residence times at the end point all have average value $E[H_i] = 1$, we can compute the steady-state probability P that the packet is at its destination point, and then obtain the average time that it takes a packet to reach the destination ET_1 using:

$$\begin{aligned} P &= \frac{1}{1 + E[T_1]} \\ E[T_1] &= \frac{1}{P} - 1. \end{aligned} \tag{4}$$

Thus the problem of calculating $E[T_1(D)]$ reduces to the problem of computing P.

A. The diffusion equation

We represent the distance of the packet to the destination by a Brownian motion [1], [2], [4]. In order to take into account the artificial holding time at the boundary $z = 0$ described above, the process Z will be represented as a Brownian motion modified to have holding times at the boundary $z = 0$, from which it jumps to an interior point (the starting point of travel at $z = D$) as suggested in some earlier work on diffusion approximations for queues [3].

The holding time at the boundary $z = 0$ corresponds to the artificial time H_i spent by the packet at $z = 0$ before it once again starts the search process. The new start of the search is represented by an instantaneous jump of the process Z from $z = 0$ to $z = D$. In addition to the holding time at $z = 0$, we also can have packet loses and time-outs. Thus in addition to the usual diffusion equation, the process we consider will have discrete (i.e. not continuous) components as described below.

At any distance $z > 0$ from the destination, the diffusion process can jump to a "loss state" L at rate $\lambda\Delta$ when a packet is lost, or it can jump to the "restart state" W at rate $r\Delta t$ if the time-out expires. From the loss state, the process will jump to the restart state at rate $r\Delta t$ because of the time-out. Finally, from the re-start state W the process will jump to the initial position of the packet at $z = D$ with a rate $m^{-1}\Delta t$.

Under these assumptions the equation governing the probability density function $f_{z,t}$ for the distance of the packet to its destination becomes:

$$\begin{aligned}\frac{\partial f_{z,t}}{\partial t} &= -(\lambda+r)f_{z,t} - b\frac{\partial f_{z,t}}{\partial z} + \frac{1}{2}c\frac{\partial^2 f_{z,t}}{\partial z^2} \\ &\quad +[P(t)+\mu W(t)\delta(z-D),\end{aligned} \tag{5}$$

where the first term on the right hand side describes the effect of the loss of the packet and the time-out, while $W(t)$ is the probability that at time t the packet is waiting to be re-sent after the time-out has been triggered.

- $P(t)$ multiplied by the rate 1 is the rate at which the packet travel process is re-started after the packet has successfully reached its destination, while
- $\mu W(t)$ is the rate at which the packet is re-sent is after a overhead delay of average value $m = \mu^{-1}$.

If $L(t)$ is the probability that the packet is in the lost state at time t, then we also have:

$$\frac{dL(t)}{dt} = -rL(t) + \lambda\int_0^\infty f_{z,t}dz, \tag{6}$$

because at any distance from the destination $z > 0$ at time t, the search packet may be lost at rate λ, while the average time spent in the lost state before a time-out occurs is r^{-1}. Similarly:

$$\frac{dW(t)}{dt} = -\mu W(t) + rL(t) + r\int_0^\infty f_{z,t}dz, \tag{7}$$

because the time-out state may be entered from the loss state or directly when the packet is in some position z during its search for the destination. Finally, we use the fact that the boundary condition at $z = 0$ is absorbing, so that

$$\frac{dP(t)}{dt} = -P(t) + \lim_{z\to 0^+}[-bf_{z,t} + \frac{1}{2}c\frac{\partial f_{z,t}}{\partial z}], \tag{8}$$

$$f_{0,t} = 0. \tag{9}$$

We must also have that the sum of the probabilities is one:

$$1 = W(t) + L(t) + P(t) + \int_0^\infty f_{z,t}dt. \tag{10}$$

B. Solving the model in steady-state

We first write the diffusion equation in steady-state:

$$-(\lambda+r)f_z - b\frac{\partial f_z}{\partial z} + \frac{1}{2}c\frac{\partial^2 f_z}{\partial z^2}, \tag{11}$$

and assuming a steady-state solution of the form:

$$f_z = Ee^{uz}, \quad z \geq D, \tag{12}$$

$$f_z = Fe^{uz} + G, \quad z \leq D \tag{13}$$

where $E,\ F,\ G$ are constants, we write the characteristic polynomial

$$0 = -(\lambda+r) - bu + \frac{1}{2}cu^2. \tag{14}$$

which has has two real roots. Since we are seeking a function $f(z)$ which is a probability density function whose integral over $[0, +\infty]$ must be finite, we take the negative root which takes a different value depending on whether b is negative or positive:

$$u = \frac{b}{c}[\sqrt{1+\frac{2(\lambda+r)c}{b^2}}+1] if\ b \leq 0 \tag{15}$$

$$u = \frac{-b}{c}[\sqrt{1+\frac{2(\lambda+r)c}{b^2}}-1] if\ b \geq 0 \tag{16}$$

Using the condition $f_0 = 0$ we have $F = -G$ and with the continuity condition for $f(z)$ at $z = D$ we obtain $E = F(1 - e^{-uD})$. Thus,

$$\int_0^\infty f_z dz = -\tfrac{E}{u}e^{uD} + \tfrac{F}{u}(e^{uD}-1) - FD = -FD.$$

Using (6) and (7) in steady state we obtain:

$$L = -\frac{\lambda}{r}FD, \tag{17}$$

$$W = -\frac{\lambda+r}{\mu}FD, \tag{18}$$

while (9) yields:

$$P = \frac{1}{2}cuF + \mu W. \tag{19}$$

Using the normalising condition for the sum of the probabilities, we get:

$$F = [\frac{1}{2}cu - D(\lambda+r) - D(1+\frac{\lambda}{r} + m(\lambda+r)]^{-1}, \tag{20}$$

and finally we have:

Result The total average time that may include several possible restarts after time-outs and retransmission overhead delays, or losses with time-outs and overhead delays, is obtained by computing $E[T] = P^{-1} - 1$ using (19) and substituting (16), (15):

$$E[T] = -D\frac{1+\frac{\lambda}{r}+m(\lambda+r)}{\frac{1}{2}cu - D(\lambda+r)}, \tag{21}$$

yielding for $R = 1/r$:

$$E[T] = \begin{cases} \frac{R+m}{1+\frac{|b|}{D(\lambda+r)}[\sqrt{1+\frac{2c(\lambda+r)}{b^2}}+1]},\ if\ b \leq 0 \\ \frac{R+m}{1+\frac{|b|}{D(\lambda+r)}[\sqrt{1+\frac{2c(\lambda+r)}{b^2}}-1]},\ if\ b > 0 \end{cases} \tag{22}$$

Since $E[T] = E[T_1(D)]$, we can express the condition (2) that relates the speed of motion of the mobile to the speed at which the packet travels if the packet is to catch up with the mobile, as:

$$D > \begin{cases} \frac{\gamma(R+m)}{1+\frac{|b|}{D(\lambda+r)}[\sqrt{1+\frac{2c(\lambda+r)}{b^2}}+1]},\ if\ b \leq 0 \\ \frac{\gamma(R+m)}{1+\frac{|b|}{D(\lambda+r)}[\sqrt{1+\frac{2c(\lambda+r)}{b^2}}-1]},\ if\ b > 0 \end{cases} \tag{23}$$

which reduces to the following compact and elegant expression:

$$D > \begin{cases} \gamma(R+m) - \frac{|b|}{\lambda+r}[\sqrt{1+\frac{2c(\lambda+r)}{b^2}}+1], \ if \ b \leq 0 \\ \gamma(R+m) - \frac{|b|}{\lambda+r}[\sqrt{1+\frac{2c(\lambda+r)}{b^2}}-1], \ if \ b > 0 \end{cases} \tag{24}$$

C. How long does it take the packet to catch the mobile?

Clearly, there will be a series of simultaneous moves by the packet and the mobile, until the mobile is within range d of the packet. The packet will move a distance D, then D_1, and so on till it moves a distance D_n such that $D_1 = E[T_1(D)]$, $D_{i+1} = E[T_1(D_i)]$, etc., and:

$$n = inf\{i : d > \gamma E[T_1(D_n)]\}, \tag{25}$$

at which point the mobile will be in hearing distance of the packet's current location. In turn, the total average travel time of the packet will have been:

$$E[T_{total}] = E[T_1(D] + \sum_{i=1}^{n} E[T_1(D_i)], \tag{26}$$

Both n and $E[T_{total}]$ will have to be iteratively computed using:

$$E[T(D_{i+1})] = \begin{cases} \frac{R+m}{1+\frac{|b|}{\gamma E[T(D_i)](\lambda+r)}[\sqrt{1+\frac{2c(\lambda+r)}{b^2}}+1]}, \ if \ b \leq 0 \\ \frac{R+m}{1+\frac{|b|}{\gamma E[T(D_i)](\lambda+r)}[\sqrt{1+\frac{2c(\lambda+r)}{b^2}}-1]}, \ if \ b > 0 \end{cases} \tag{27}$$

with $D_0 = D$.

IV. Optimising the Time-Out

Clearly, the faster the packet can travel, the more quickly it will reach the mobile. In this section we will see how the time-out rate r can be used to minimise the travel time.

The case where $b < 0$ can be viewed as the "normal" or favourable situation when the packet typically makes some headway towards its destination each time it is transmitted. In this case, equation (21) can be viewed as the sum of two positive terms. The first of them:

$$\tau_1(r) = \frac{\frac{1}{r}}{1 - \frac{b}{D(\lambda+r)}[1+\sqrt{1+\frac{2(\lambda+r)c}{b^2}}]}, \tag{28}$$

is monotone decreasing in r and tends to zero as r tends to infinity. The second term:

$$\tau_2(r) = \frac{m}{1 - \frac{b}{D(\lambda+r)}[1+\sqrt{1+\frac{2(\lambda+r)c}{b^2}}]}, \tag{29}$$

is monotone increasing with r and tends to m as r tends to infinity. Clearly, there will be some value of r, say r^*, which optimises the value of $E[T]$. It can also be seen that the second derivative of $E[T]$ is positive, so that this value r^* is a minimum.

Figure 1 shows how the average time $E[T]$ is strongly influenced by the time-out and by the variance parameter c. We see that when packet loss can occur (in this case $\lambda = 0.1$), increasing variance parameter c actually reduces the average search time. This is because losses are followed by re-starts of the search, and a larger variance with the same average value increases the chance of finding a shorter path. We also see that there is definitely an optimum value of the time-out (in this case that minimises $E[T]$.

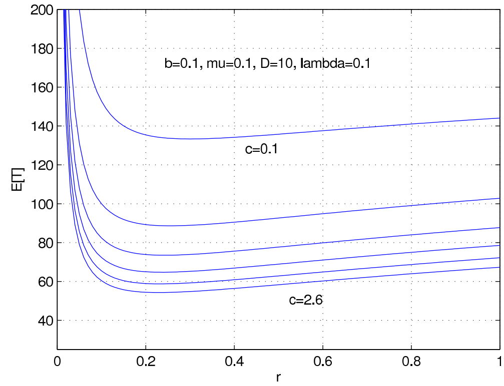

Fig. 1. The effect of varying the time-out rate r, for D=10, an average value of the overhead time for packet retransmission of $m = 10$, loss rate lambda=0.1 and different values of the variance parameter c=0.1, 0.6, 1.1, 1.6, 2.1, 2.6

The next Figure 2 illustrates the effect of the packet loss rate $\lambda = 0.1$ to $\lambda = 0.5$ for fixed c and varying r. It shows that as the loss rate increases, so does the average time $E[T]$ for the packet to reach its destination, as expected, if all other parameters remain the same. Again we see that the average value of the time-out will significantly influence the time it takes the packet to reach its destination and that an optimum value can be obtained.

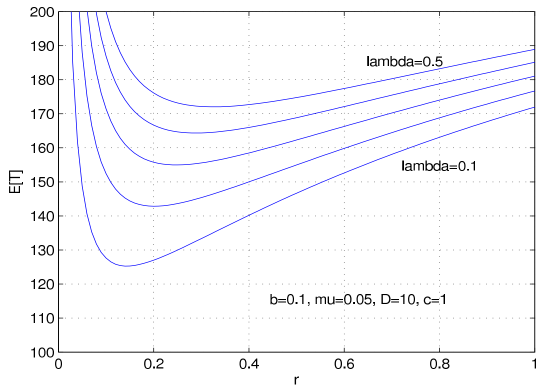

Fig. 2. The effect of varying the time-out rate r, with b=0.1, D=10, average value of the overhead before packet retransmission of $m = 20$, and values of the loss rate ranging from 0.1 to 0.5 in steps of 0.1

A. Perfect Ignorance or b=0

An interesting and surprising result occurs for $b = 0$. This corresponds to the case wehere the packet is being routed under *perfect ignorance*: at each step it neither gets further away nor closer to the destination. We see that if packet loss can occur ($\lambda > 0$) or if time-outs exist ($r > 0$), the travel time of the packet to the destination remains finite on the average if $c > 0$ and is given by:

$$E[T_{b=0}] = D\frac{1 + \frac{\lambda}{r} + m(\lambda + r)}{\sqrt{2(\lambda + r)c} + D(\lambda + r)} \quad (30)$$

Thus even though we have a "stupid search" for the destination, without a clear direction which way to go, the packet does make it in a finite amount of time on the average. The numerical results of Figure 3 tell us that when the variance of the change in position per unit time c increases, the time to reach the destination decreases. The last Figure 4 shows how,

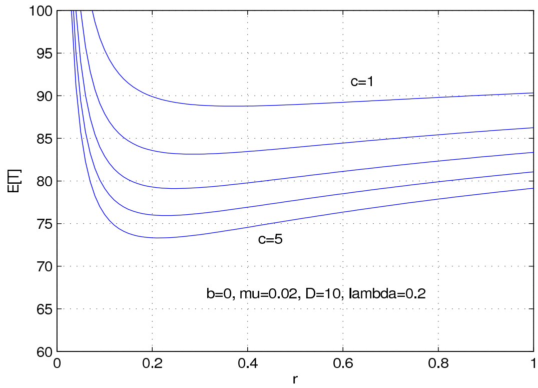

Fig. 3. The effect of varying the time-out rate r, with b=0, loss rate lambda=0.2, D=10, average value of the overhead before packet retransmission of $m = 50$, and values of the variance parameter c ranging from 1 to 5

for a fixed value of c, the loss rate influences the average time $E[T]$. Again we see that there is an optimum value of the average time-out $1/r$ which results in the smallest value of $E[T]$.

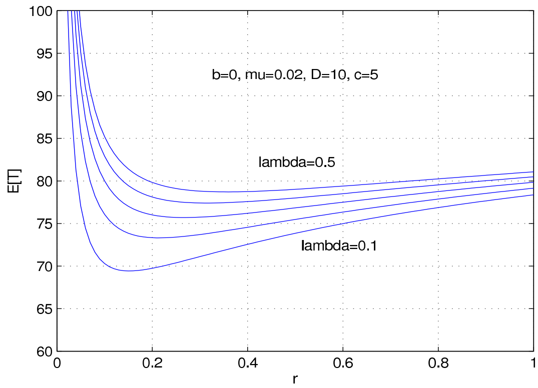

Fig. 4. The effect of varying the time-out rate r, with b=0, D=10, average value of the overhead before packet retransmission of $m = 50$, and loss rate values 0.1, 0.2, 0.3, 0.4, 0.5

V. Conclusions

In this paper we derive closed form results for the average travel time of a packet in a homogenous multi-hop medium as a function of the distance between the source and the destination, using a continuous space approximation. The effect of packet loss and of time-outs is also considered. We show that if that the time-out interval can be selected so as to minimise the average travel time, and that even when the packet will on the average not make any progress to its destination, the use of a time-out with repeated attempts to send the packet can insure that the packet eventually makes it to the destination in finite average time.

We also address the case where the destination node moves, and derive conditions that must be satisfied so that the packet is able to attain the mobile node despite the motion of the destination node. We also suggest an iterative procedure to estimate the total travel when the destination node moves.

References

[1] A. Einstein. Investigations on the Theory of Brownian Motion. Dutton, New York, 1926, reprinted by Dover, New York, 1956.

[2] W. Feller. An Introduction to Probability Theory and its Applications, Vols. I and II. Wiley, New York, 1966.

[3] E. Gelenbe. "On approximate computer system models", *Journal ACM*, 22 (2), pp. 261-269, April 1975.

[4] J. Medhi. Stochastic Models in Queueing Theory. pp. 373 ff. Academic Press, New York, 1991.

[5] Young-Bae Ko and N. H. Vaidya. "Location-aided routing (lar) in mobile ad hoc networks", In *Proc. ACM MOBICOM*, pp. 66–75, 1998.

[6] E. Royer and C-K. Toh. "A review of current routing protocols for ad-hoc mobile wireless networks", *IEEE Pesonal Communications* 6 (2), pp. 46–55, 1999.

[7] C. E. Perkins (ed.) Ad Hoc Networking. Adisson-Wesley, Longman, 2000.

[8] C. Intanagonwiwowat, R. Govindan and D. Estrin. "Directed diffusion: a scalable and robust communication paradigm for sensor networks", *Proc. 6th Annual International Conf. on Mobile Computing and Networks ACM MOBICOM 2000*, pp. 56–67, 2000.

[9] B. Krishnamachari, D. Estrin and S. Wicker. "Modelling data-centric routing in wireless sensor networks", *IEEE INFOCOM*, 2002.

[10] I.F. Akyildiz, W. Su, Y. Sankarasubramanian and E. Çayırcıi. "A survey on sensor networks", *IEEE Comm. Magazine*, pp. 102–114, August 2002.

[11] M. Zorzi, R.R. Rao, "Geographic random forwarding (GeRaF) for ad hoc and sensor networks: multihop performance", *IEEE Transactions on Mobile Computing*, 2 (4): 337–348, 2003.

[12] S. Shakkottai, R. Srikant, N. Shroff, "Unreliable sensor grids: coverage, connectivity and diameter", *Proc. 22nd INFOCOM Annual Joint Conference of the IEEE Computer and Communications Societies* Vol. 2: 1073–1083, 2003.

[13] H. M. F. Abo El Fotoh, S. S. Iynegar, K. Chakrabarty. "Computing reliability and message delay for cooperative wireless distributed sensor networks subject to random failures", *IEEE Transactions on Reliability* 54 (1), March 2005.

[14] E. Gelenbe, M. Gellman, R. Lent, P. Liu, Pu Su. "Autonomous smart routing for network QoS", *Proc. First International Conference on Autonomic Computing*, (IEEE Computer Society), ISBN 0-7695-2114-2, pp. 232-239, May 17-18, 2004, New York.