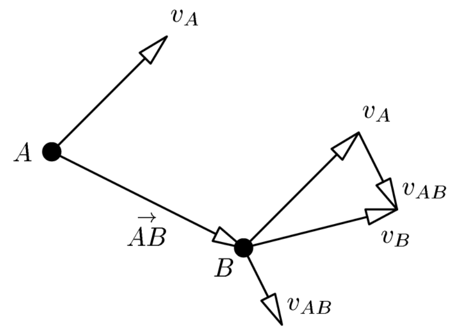
Note: This pdf document you are reading can be produced from the LaTeX source file `README.tex`. You need to have `pdflatex` installed. If your system has `make` installed, simply typing `make README.pdf` will produce `README.pdf`.

All the included files are generated on a Linux machine and Windows users may encounter some trouble opening the files.

1 Derivative of the distance between two nodes



$$
\begin{array}{lcl}
\text{node } A : (x_1, y_1) & \longrightarrow & \text{velocity } v_A : (\dot{x}_1, \dot{y}_1) \\
\text{node } B : (x_2, y_2) & \longrightarrow & \text{velocity } v_B : (\dot{x}_2, \dot{y}_2)
\end{array}
$$

$$
\begin{aligned}
d_{AB} &= \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} = \text{distance between } A \text{ and } B \\
v_{AB} &= v_B - v_A
\end{aligned}
$$

Note that $\frac{d}{dt}d_{AB}$, the time derivative of the distance d_{AB}, is a **scalar**, while v_{AB} is a **a vector**. Since a motion orthogonal to the vector $\vec{AB}$ does not cause any distance change between nodes A and B, from the figure above:

$$
\begin{aligned}
\frac{d}{dt}d_{AB} &= v_{AB} \cdot u_{AB} \qquad \text{(an inner product)} \\
&= (\dot{x}_2 - \dot{x}_1, \dot{y}_2 - \dot{y}_1) \cdot \frac{(x_2 - x_1, y_2 - y_1)}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}} \\
&= \frac{1}{d_{AB}}\Big((\dot{x}_2 - \dot{x}_1)(x_2 - x_1) + (\dot{y}_2 - \dot{y}_1)(y_2 - y_1)\Big),
\end{aligned}
\tag{1}
$$

where

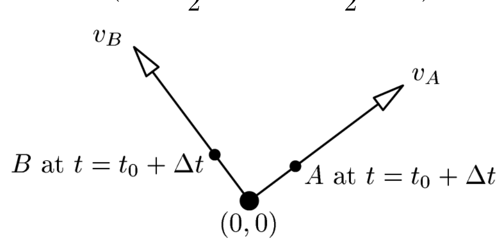
$$
u_{AB} = \frac{\vec{AB}}{|\vec{AB}|} = \frac{\vec{AB}}{d_{AB}}
$$

is a unit vector in direction $\vec{AB}$. The same result can also be obtained by a simple derivative as follows:

$$
\frac{d}{dt}d_{AB} = \frac{d}{dt}\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}.
$$

However, if $d_{AB} \to 0$, we have a problem! If $v_A = v_B$, we have a trivial solution $\frac{d}{dt}d_{AB} = 0$. Assume $v_A \neq v_B$. Without loss of generality, assume node A and B are at the origin at $t = t_0$, and the velocity of nodes A and B are v_A and v_B, respectively. For arbitrarily small $\Delta t > 0$, at $t = t_0 + \Delta t$, the velocities and the locations of nodes A and B are

$$
\begin{aligned}
v_A &= (\dot{x}_1 + \ddot{x}_1\Delta t, \dot{y}_1 + \ddot{y}_1\Delta t) \\
v_B &= (\dot{x}_2 + \ddot{x}_2\Delta t, \dot{y}_2 + \ddot{y}_2\Delta t) \\
A &= \left(\, \dot{x}_1 + \frac{1}{2}\ddot{x}_1\Delta t \,,\, \dot{y}_1 + \frac{1}{2}\ddot{y}_1\Delta t \,\right) \Delta t \\
B &= \left(\, \dot{x}_2 + \frac{1}{2}\ddot{x}_2\Delta t \,,\, \dot{y}_2 + \frac{1}{2}\ddot{y}_2\Delta t \,\right) \Delta t
\end{aligned}
$$



From equation (1),

$$
\begin{aligned}
\frac{d}{dt}d_{AB} &= \lim_{\Delta t\to 0}\big(\,\dot{x}_2+\ddot{x}_2\Delta t-\dot{x}_1-\ddot{x}_1\Delta t\,,\,\dot{y}_2+\ddot{y}_2\Delta t-\dot{y}_1-\ddot{y}_1\Delta t\,\big)\\
&\qquad\cdot\frac{\big(\dot{x}_2+\frac{1}{2}\ddot{x}_2\Delta t-\dot{x}_1-\frac{1}{2}\ddot{x}_1\Delta t\,,\,\dot{y}_2+\frac{1}{2}\ddot{y}_2\Delta t-\dot{y}_1-\frac{1}{2}\ddot{y}_1\Delta t\big)}{\sqrt{\big(\dot{x}_2+\frac{1}{2}\ddot{x}_2\Delta t-\dot{x}_1-\frac{1}{2}\ddot{x}_1\Delta t\big)^2\,+\,\big(\dot{y}_2+\frac{1}{2}\ddot{y}_2\Delta t-\dot{y}_1-\frac{1}{2}\ddot{y}_1\Delta t\big)^2}} \qquad (2)\\
&= (\dot{x}_2-\dot{x}_1,\dot{y}_2-\dot{y}_1)\cdot\frac{(\dot{x}_2-\dot{x}_1,\dot{y}_2-\dot{y}_1)}{\sqrt{(\dot{x}_2-\dot{x}_1)^2+(\dot{y}_2-\dot{y}_1)^2}}\\
&= \sqrt{(\dot{x}_2-\dot{x}_1)^2+(\dot{y}_2-\dot{y}_1)^2}.
\end{aligned}
$$

If $\dot{x}_2=\dot{x}_1$ but $\dot{y}_2\neq\dot{y}_1$ then

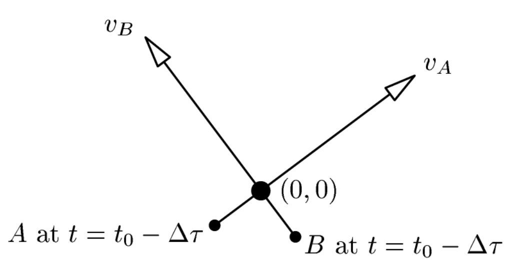
$$\frac{d}{dt}d_{AB}=\mid\dot{y}_2-\dot{y}_1\mid,$$

if $\dot{x}_2\neq\dot{x}_1$ but $\dot{y}_2=\dot{y}_1$ then

$$\frac{d}{dt}d_{AB}=\mid\dot{x}_2-\dot{x}_1\mid.$$

But, if we choose $\Delta t=-\Delta\tau<0$, then at $t=t_0+\Delta t=t_0-\Delta\tau$,

$$
\begin{aligned}
v_A &= (\dot{x}_1-\ddot{x}_1\Delta\tau,\dot{y}_1-\ddot{y}_1\Delta\tau)\\
v_B &= (\dot{x}_2-\ddot{x}_2\Delta\tau,\dot{y}_2-\ddot{y}_2\Delta\tau)\\
A &= -\big(\,\dot{x}_1-\frac{1}{2}\ddot{x}_1\Delta\tau\,,\,\dot{y}_1-\frac{1}{2}\ddot{y}_1\Delta\tau\,\big)\,\Delta\tau\\
B &= -\big(\,\dot{x}_2-\frac{1}{2}\ddot{x}_2\Delta\tau\,,\,\dot{y}_2-\frac{1}{2}\ddot{y}_2\Delta\tau\,\big)\,\Delta\tau
\end{aligned}
$$



Then

$$
\begin{aligned}
\frac{d}{dt}d_{AB} &= \lim_{\Delta\tau\to 0}\big(\,\dot{x}_2-\ddot{x}_2\Delta\tau-\dot{x}_1+\ddot{x}_1\Delta\tau\,,\,\dot{y}_2-\ddot{y}_2\Delta\tau-\dot{y}_1+\ddot{y}_1\Delta\tau\,\big)\\
&\qquad\cdot\frac{\big(-\dot{x}_2+\frac{1}{2}\ddot{x}_2\Delta\tau+\dot{x}_1-\frac{1}{2}\ddot{x}_1\Delta\tau\,,\,-\dot{y}_2+\frac{1}{2}\ddot{y}_2\Delta\tau+\dot{y}_1-\frac{1}{2}\ddot{y}_1\Delta\tau\big)}{\sqrt{\big(-\dot{x}_2+\frac{1}{2}\ddot{x}_2\Delta\tau+\dot{x}_1-\frac{1}{2}\ddot{x}_1\Delta\tau\big)^2\,+\,\big(-\dot{y}_2+\frac{1}{2}\ddot{y}_2\Delta\tau+\dot{y}_1-\frac{1}{2}\ddot{y}_1\Delta\tau\big)^2}}\\
&= -(\dot{x}_2-\dot{x}_1,\dot{y}_2-\dot{y}_1)\cdot\frac{(\dot{x}_2-\dot{x}_1,\dot{y}_2-\dot{y}_1)}{\sqrt{(\dot{x}_2-\dot{x}_1)^2+(\dot{y}_2-\dot{y}_1)^2}}\\
&= -\sqrt{(\dot{x}_2-\dot{x}_1)^2+(\dot{y}_2-\dot{y}_1)^2}.
\end{aligned}
$$

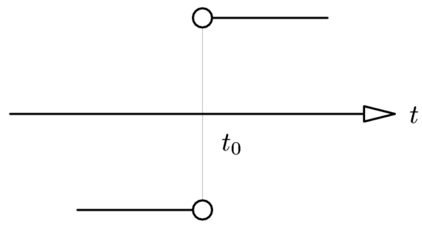
If $\dot{x}_2=\dot{x}_1$ but $\dot{y}_2\neq\dot{y}_1$ then

$$\frac{d}{dt}d_{AB}=-\mid\dot{y}_2-\dot{y}_1\mid,$$

if $\dot{x}_2\neq\dot{x}_1$ but $\dot{y}_2=\dot{y}_1$ then

$$\frac{d}{dt}d_{AB}=-\mid\dot{x}_2-\dot{x}_1\mid.$$

Therefore, when two nodes meet at $t=t_0$, the derivative of the distance between the nodes can be illustrated by the following figure:

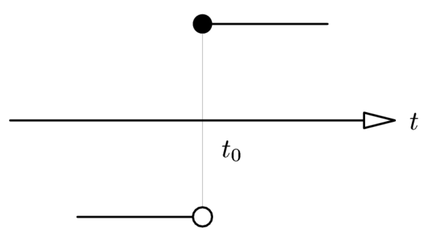


That is, if A and B meet at $t = t_0$

$$\begin{aligned} \frac{d}{dt} d_{AB} < 0 & \qquad \text{for } t < t_0; \\ \frac{d}{dt} d_{AB} > 0 & \qquad \text{for } t > t_0; \\ \text{undefined} & \qquad \text{at } t = t_0. \end{aligned}$$

In our simulation, we choose

$$\begin{aligned} \frac{d}{dt} d_{AB} < 0 & \qquad \text{for } t < t_0; \\ \frac{d}{dt} d_{AB} \geq 0 & \qquad \text{for } t \geq t_0. \end{aligned}$$

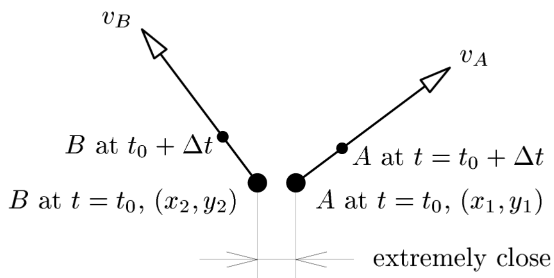


Note: The time derivative of $\frac{d}{dt} d_{AB}$ is a discontinuous function of time where the discontinuities occur when two nodes meet in space. It is not an error if your code generates a discontinuous function!

Pseudo code 1

If node A and B are the same nodes, then
    return 0
else
    if node A and B are NOT at the same location
        return $v_{AB} \cdot \frac{\vec{AB}}{|\vec{AB}|}$
    else (i.e., A and B are at the same location)
        if $v_A = v_B$ then
            return 0
        else
            return $\sqrt{(\dot{x}_2 - \dot{x}_1)^2 + (\dot{y}_2 - \dot{y}_1)^2}$

In reality, the scenario described above does not occur (it happens with probability 0). It is more likely that two nodes can get close but do not actually meet. However, if two nodes get very close, a numerical problem can occur.



Assume A and B are extremely close at $t = t_0$, then using (1) will cause a numerical instability due to the division by d_{AB}. At time $t = t_0$

$$\begin{aligned} v_A &= (\dot{x}_1, \dot{y}_1) \\ v_B &= (\dot{x}_2, \dot{y}_2) \\ A = (x_1, y_1) \; &\approx \; (x_2, y_2) = B \end{aligned}$$

and at time $t = t_0 + \Delta t$

$$\begin{aligned} v_A &= (\dot{x}_1 + \ddot{x}_1 \Delta t \, , \, \dot{y}_1 + \ddot{y}_1 \Delta t) \\ v_B &= (\dot{x}_2 + \ddot{x}_2 \Delta t \, , \, \dot{y}_2 + \ddot{y}_2 \Delta t) \\ A &= \Big(x_1 + (\dot{x}_1 + \frac{\ddot{x}_1 \Delta t}{2}) \, \Delta t \; , \; y_1 + (\dot{y}_1 + \frac{\ddot{y}_1 \Delta t}{2}) \, \Delta t \Big) \\ B &= \Big(x_2 + (\dot{x}_2 + \frac{\ddot{x}_2 \Delta t}{2}) \, \Delta t \; , \; y_2 + (\dot{y}_2 + \frac{\ddot{y}_2 \Delta t}{2}) \, \Delta t \Big) \end{aligned}$$

To avoid a numerical instability, we can use an approximation of (2) by substituting small Δt instead of taking an infinite limit of Δt:

$$\begin{aligned} \frac{d}{dt} d_{AB} \approx & \left(\dot{x}_2 + \ddot{x}_2 \Delta t - \dot{x}_1 - \ddot{x}_1 \Delta t \, , \, \dot{y}_2 + \ddot{y}_2 \Delta t - \dot{y}_1 - \ddot{y}_1 \Delta t \right) \\ & \cdot \frac{\left(x_2 + (\dot{x}_2 + \frac{\ddot{x}_2 \Delta t}{2}) \Delta t - x_1 - (\dot{x}_1 + \frac{\ddot{x}_1 \Delta t}{2}) \Delta t \, , \; y_2 + (\dot{y}_2 + \frac{\ddot{y}_2 \Delta t}{2}) \Delta t - y_1 - (\dot{y}_1 + \frac{\ddot{y}_1 \Delta t}{2}) \Delta t \right)}{\sqrt{\left(x_2 + (\dot{x}_2 + \frac{\ddot{x}_2 \Delta t}{2}) \Delta t - x_1 - (\dot{x}_1 + \frac{\ddot{x}_1 \Delta t}{2}) \Delta t \right)^2 \; + \; \left(y_2 + (\dot{y}_2 + \frac{\ddot{y}_2 \Delta t}{2}) \Delta t - y_1 - (\dot{y}_1 + \frac{\ddot{y}_1 \Delta t}{2}) \Delta t \right)^2}} \end{aligned}$$

Since Δt is very small, it is reasonable to assume that $\ddot{x}_i \approx 0$ and $\ddot{y}_i \approx 0$ for $i = 1, 2$ (i.e., the velocity remains unchanged). Since $(x_1, y_1) \; \approx \; (x_2, y_2)$, the above equation can be further approximated as follows.

$$\begin{aligned} \frac{d}{dt} d_{AB} &\approx (\, \dot{x}_2 - \dot{x}_1 \, , \, \dot{y}_2 - \dot{y}_1 \,) \cdot \frac{\left((\dot{x}_2 - \dot{x}_1) \Delta t \; , \; (\dot{y}_2 - \dot{y}_1) \Delta t \right)}{\sqrt{(\dot{x}_2 - \dot{x}_1)^2 (\Delta t)^2 \; + \; (\dot{y}_2 - \dot{y}_1)^2 (\Delta t)^2}} \\ &= (\, \dot{x}_2 - \dot{x}_1 \, , \, \dot{y}_2 - \dot{y}_1 \,) \cdot \frac{(\, \dot{x}_2 - \dot{x}_1 \, , \, \dot{y}_2 - \dot{y}_1 \,)}{\sqrt{(\dot{x}_2 - \dot{x}_1)^2 \; + \; (\dot{y}_2 - \dot{y}_1)^2}} \\ &= \sqrt{(\dot{x}_2 - \dot{x}_1)^2 \; + \; (\dot{y}_2 - \dot{y}_1)^2} \end{aligned}$$

Pseudo code 2

If node A and B are the same nodes, then
    return 0
else
    if $d_{AB} > \epsilon_d$
        return $v_{AB} \cdot \frac{\vec{AB}}{|\vec{AB}|}$
    else (i.e., $d_{AB} \le \epsilon_d$)
        if $\mid v_B - v_A \mid \le \epsilon_v$ then
            return 0
        else
            return $\sqrt{(\dot{x}_2 - \dot{x}_1)^2 + (\dot{y}_2 - \dot{y}_1)^2}$

The above pseudo code 2 can be simplified as follows:

Pseudo code 3 (final)

If node A and B are the same nodes, then
  return 0
else
  if $d_{AB} > \epsilon$
    return $v_{AB} \cdot \frac{\vec{AB}}{|\vec{AB}|}$
  else (i.e., $d_{AB} \leq \epsilon$)
    return $\sqrt{(\dot{x}_2 - \dot{x}_1)^2 + (\dot{y}_2 - \dot{y}_1)^2}$

2 Mobility Models

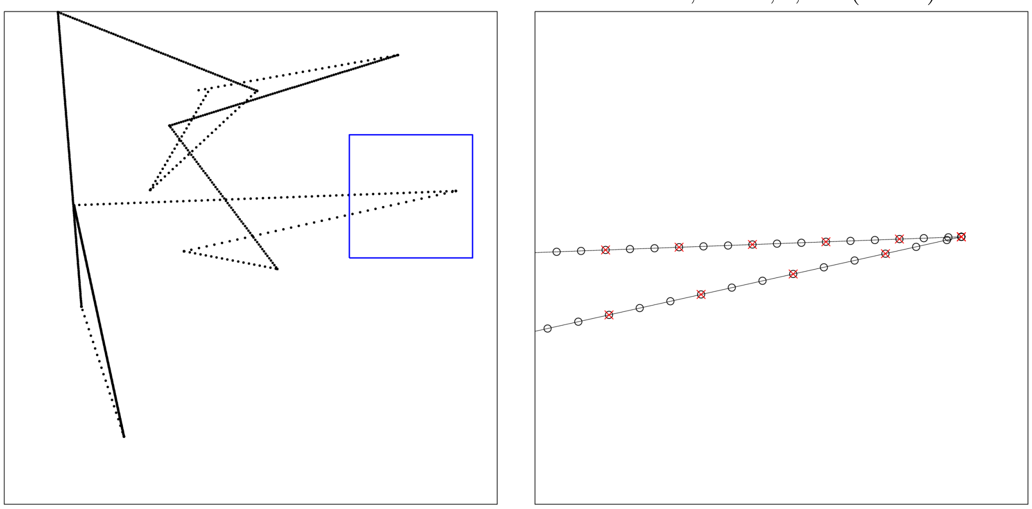
2.1 Random Waypoint Model

In the random waypoint (RWP) model, a node selects a random destination uniformly distributed over a predefined region and moves to the destination at a random speed uniformly distributed between the minimum and maximum speed. Reaching the destination, after pausing for a certain period of time, the node selects a new random destination and speed.

A typical trajectory of a node moving in Random Waypoint model is shown in the figure below, where the right hand side figure shows the details of the region marked by a blue square.

Each symbol represents the location of a node:

at $t = 0.1k,\ k = 0, 1, \cdots$ | at $t = 0.1k,\ k = 0, 1, \cdots$ (black circle)
at $t = 0.3k,\ k = 0, 1, \cdots$ (red $\times$)



Network dimension : (6×6)
distribution of the random speed : $U[0.1, 1]$
pause time : 1 sec.
Note: All figures normalized by the communication range.

For more information, see

Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, and Jorjeta Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols," *Proc. IEEE/ACM Mobicom'98,* Oct. 1998.

2.2 Random Gauss-Markov Model

In the random Gauss-Markov (RGM) model, each node is assigned a speed v and direction θ, and v and θ are updated every Δt as follows:

$$v(t + \Delta t) = \min[\max(v(t) + \Delta v, V_{\min}), V_{\max}],$$
$$\theta(t + \Delta t) = \theta(t) + \Delta\theta,$$

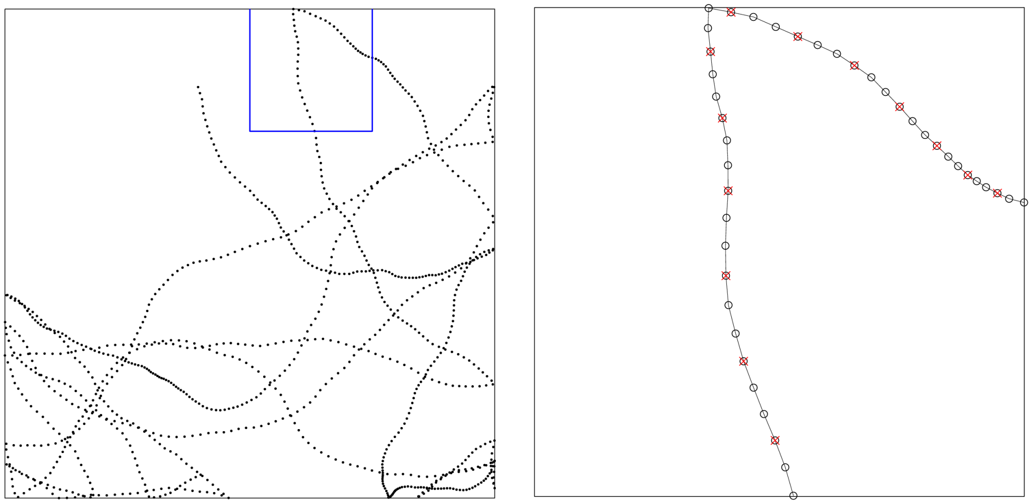
where $V_{\min}$ and $V_{\max}$ are the minimum and maximum speed of the node, and Δv and $\Delta\theta$ are random variables with uniform distribution over the intervals $[-\Delta v_{\max}, \Delta v_{\max}]$ and $[-\Delta\theta_{\max}, \Delta\theta_{\max}]$, respectively. When a node reaches a boundary, the node reflects off the boundary by choosing a new random direction. However, the updates of the v and θ can be implemented in various ways. The following example implements the model introduced in the Master's thesis by Shukla. For another example of the implementation of the RGM model, see the paper by Camp et al.

A typical trajectory of a node moving in Random Gauss-Markov model is shown in the figure below, where the right hand side figure shows the details of the region marked by a blue square.

Each symbol represents the location of a node:

at $t = 0.1k$, $k = 0, 1, \cdots$

at $t = 0.1k$, $k = 0, 1, \cdots$ (black circle)
at $t = 0.3k$, $k = 0, 1, \cdots$ (red $\times$)



Network dimension : (6×6)
minimum node speed : 0.0
maximum node speed : 1.0
direction/speed update interval : 0.2 sec.
Δv : $U[-0.1, 0.1]$
$\Delta \theta$: $U[-0.1\pi, 0.1\pi]$
Note: All figures normalized by the communication range.

For more information, see

Deepanshu Shukla, "Mobility Models in ad hoc networks," Master's thesis, KReSIT-ITT Bombay, Nov. 2001.

Tracy Camp, Jeff Boleng, and Vanessa Davies, "A Survey of Mobility for Ad Hoc Network Research," *Wireless Communication & Mobile Computing (WCMC),* Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications, 2002.

2.3 Reference Point Group Mobility Model

The implementation of the Reference Point Group Mobility (RPGM) model should be straightforward from the examples shown above. For more information about RPGM, see

Byung-Jae Kwak, Nah-Oak Song, and L. E. Miller, "A Standard Measure of Mobility for Evaluating Mobile Ad Hoc Network Performance," *IEICE Trans. Communication,* vol.E86-B, no. 11, Nov. 2003.

3 Mobility Measure

The mobility measure proposed in the paper is defined as follows.

$$M(t) = \frac{1}{N} \sum_{i=0}^{N-1} M_i(t),$$

where N is the number of nodes and

$$M_i(t) = \frac{1}{N-1} \sum_{j=0}^{N-1} \left| \frac{d}{dt} F(d_{ij}(t)) \right|.$$

$M_i(t)$ is a measure of the relative movement of other nodes as seen by node i. In the paper (and in the C code), we use notations $M^G(t)$ and $M^I(t)$ to represent the mobility measure with certain remoteness function ('G' for Gamma, 'I' for identity).

Note that the proposed mobility measure is a function of time t. In a steady state network, the time average of the measure is use for more reliable estimation of the mobility. However, for a network not in steady state, the mobility measure as a function of time can be used to investigate a certain event at certain time instance.

4 Single-hop link matrix

In the simulation, the link change rate is calculate by a single-hop link matrix and counting the changes of the elements. If there are N nodes, the single-hop link matrix will be an $(N \times N)$ matrix of integer values. If there is a single hop link between node i and node j, the (i, j)-th element of the matrix has value 1. If there is no single hop link between node i and node j, the (i, j)-th element of the matrix has value -1. Zero means zero hops, so all (i, i)-th element of the matrix has zero.

Please take a look at the function `RWP_node_single_hop_links()` in the file `function.c` for more information.

5 Included Programs

`way_point_mobility_model.c`

This program show a typical trajectory of a node moving in **Random Waypoint** mobility model.

A single node is moving in a rectangular region defined by [`X_MIN`, `X_MAX`, `Y_MIN`, `Y_MAX`].

The data for the figures in Section 2.1 are obtained using this program.

Bug: In Random Waypoint model, of the random velocity happens to be zero, the node will never be able to reach the destination. Furthermore, a zero velocity causes a "divide by zero" problem. Thus, in our simulation, the random velocity has a uniform distribution of [`V_MIN`, `V_MAX`], where `V_MIN` > 0.

`random_gauss_markov_model.c`

This program show a typical trajectory of a node moving in **Random Gauss-Markov** mobility model.

A single node is moving in a rectangular region defined by [`X_MIN`, `X_MAX`, `Y_MIN`, `Y_MAX`].

The data for the figures in Section 2.2 are obtained using this program.

`mobility_measure_RWP.c`

With mobile nodes moving in the **Random Waypoint** mobility model, this program calculates the mobility measure, and the link change rate. The Gamma function and identity functions are considered as remoteness functions in the mobility measure.

This program creates the following files:
`RWP.log`
`RWP_nodexxx.trace`, where `xxx` = `000`, `001`, $\cdots$, `039`.

A sample run result, and the parameters used are as follows.

`X_MIN`	0.000000	`Y_MIN`	0.000000
`X_MAX`	6.000000	`Y_MAX`	6.000000
`V_MIN`	0.100000	`V_MAX`	1.000000
Nunber of nodes	40	Pause time	4.000000
(In gamma function) r	5.000000	M^I	0.285408
normalized link changes rate	0.055205	M^G	0.059631

The mobility measure $M^G = 0.059631$ and the normalized link change rate $= 0.055205$ corresponds to the point S7 in Fig.5(b) in the paper:

Byung-Jae Kwak, Nah-Oak Song, and L. E. Miller, "A Standard Measure of Mobility for Evaluating Mobile Ad Hoc Network Performance," *IEICE Trans. Communication,* vol.E86-B, no. 11, Nov. 2003.

`mobility_measure_RGM.c`

This program is the same as `mobility_measure_RWP.c`, except that this one uses **Random Gauss-Markov** mobility model instead of Random Waypoint mobility model.

This program creates the following files:
`RGM.log`
`RGM_nodexxx.trace`, where `xxx` = `000`, `001`, $\cdots$, `039`.

A sample run result, and the parameters used are as follows.

`X_MIN`	0.000000	`X_MAX`	6.000000
`Y_MIN`	0.000000	`Y_MAX`	6.000000
`V_MIN`	0.100000	`V_MAX`	1.000000
Nunber of nodes	40	Δt	0.2 sec.
Δv	0.1	$\Delta\theta$	0.1π
(In gamma function) r	5.000000	M^I	0.471999
normalized link change rate	0.062154	M^G	0.068729

The mobility measure $M^G = 0.068729$ and the normalized link change rate $= 0.062154$ corresponds to the point T2 in Fig.5(b).

5.1 Files

functions.c
functions.h
Makefile
mobility_measure_RGM.c
mobility_measure_RWP.c
node_types.h
random_gauss_markov_model.c
way_point_mobility_model.c

Note: The function `gamma_pdf()` defined in the file `functions.c` uses a function named `lgamma()` provided in the GNU C Library. `lgamma(double x)` returns the natural logarithm of the absolute value

$$\int_0^\infty t^{x-1}e^{-t}dt.$$

The sign of the function is stored in the global variable `signgam`, which is declared in `math.h`. For more information of the function, please read the reference manual available at `www.gnu.org`. If you cannot use the GNU C Library for some reason, you can create your own gamma function (see Section 6.1, Numerical Recipes in C, 2nd Ed.).

Warning: The programs are not optimized for speed.

6 Mobility Measure vs. Link Change Rate

Most performance studies of the routing protocols for MANET use very simple model in their simulations. In such simple models, link change rate can be used quite effectively as a mobility measure. However, as the simulation models gets complicated, using link change rate as a mobility measure is not a feasible solution.

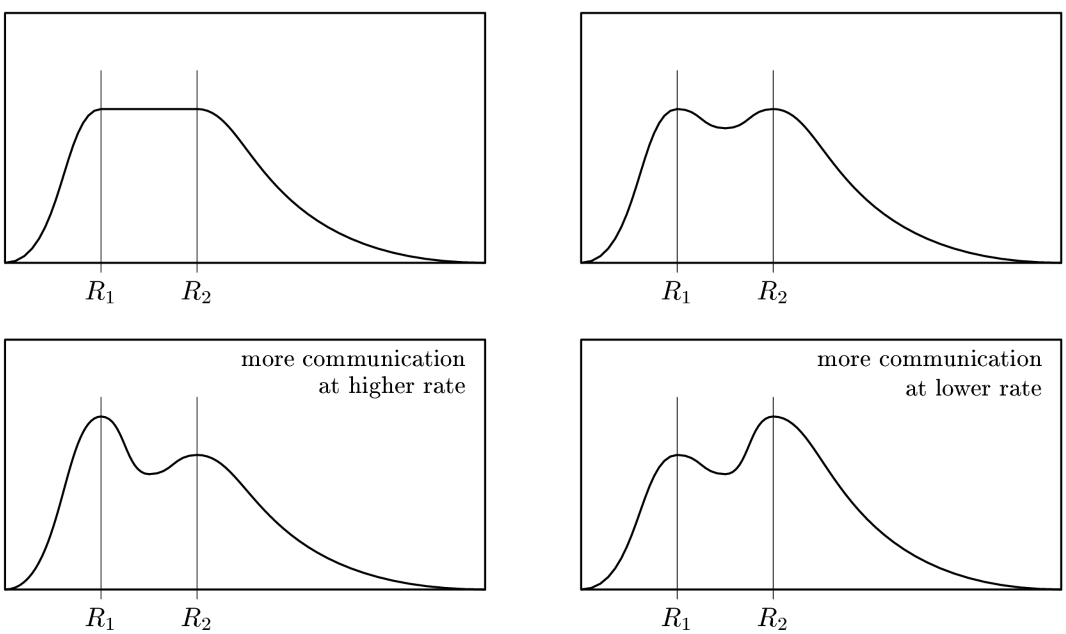
Many wireless communication links support multiple transmission rates using different modulation schemes. Consider a wireless communication link with two transmission rate (link capacity) L_1 and L_2 as follows:

	Modulation scheme 1		Modulation scheme 2
transmission rate	L_1	$>$	L_2
communication range	R_1	$<$	R_2

Assume two nodes are moving away from each other and the distance between them at time $t = t_1$ is D_1 ($< R_1$), and the distance between them at time $t = t_2 > t_1$ is D_2 and $R_1 < D_2 < R_2$. At $t = t_2$, there are three possible scenarios in terms of link status between the two nodes:

- ✓ **Broken** If the original link (at time $t = t_1$) was established at transmission rate L_1 and the service provided by the link cannot be supported by transmission rate L_2.
- ✓ **Alive but status changed** If the original link was established at transmission rate L_1 but switched to a new rate L_2 due to the distance change.
- ✓ **Alive with no change** If the original link was established at transmission rate L_2.

How to deal with the link status changes in a MANET is a problem that needs to be addressed by the routing protocol. However, more link status change means more routing overhead and it should be addressed in the measure of the mobility. Obviously, using the link change rate as a mobility measure is not possible in this example. One of the two strengths of the proposed mobility measure is the flexibility and it comes from the comes from the flexibility of the remoteness function. Some of the possible examples of the remoteness function for the scenario illustrated above are as follows (Note: The following figures represent the **derivatives** of the remoteness functions.):



Remarks In real wireless communication links, the transmission range and the carrier sense range are different, and it causes hidden and exposed terminal problems. In a multi-hop wireless communication environment such as MANET, the hidden and exposed terminal problems play more significant role compared to a single-hop wireless communication environment such as WLAN. The remoteness function can be designed to take this into account, but it requres good understanding of the effect of the hidden and exposed terminal problems.