# Jitter Considerations in On-demand Route Discovery for Mobile Ad Hoc Networks

Jiazi Yi
LIX, École Polytechnique
France
jiazi@jiaziyi.com

Juan Antonio Cordero Fuertes
ICTEAM, Université catholique de Louvain
Belgium
j.a.cordero@gmail.com

Thomas Clausen
LIX, École Polytechnique
France
thomas@thomasclausen.org

*Abstract*—**Jittering (a small, random variation in timing of control message emission) is widely used in protocols for wireless communication, in order to avoid simultaneous packet transmissions over the same channel by adjacent nodes in the network. Used for both regularly scheduled packets, for event-triggered packets, and for scheduled resets in the network, jittering is a particularly important mechanism when a network event may cause multiple adjacent nodes to react concurrently. Introduced in the proactive MANET routing protocol OLSR, the "LLN On-demand Ad hoc Distance-vector Routing Protocol - Next Generation" (LOADng), a derivative of AODV, is specified so as to also use jitter for flooding Route Request (RREQ) messages during route discovery. This use of jitter in RREQ flooding is, however, not without drawbacks, which are identified and addressed in this paper within the framework of a more general study of jitter mechanisms used for route discovery in reactive routing protocols. The paper studies the behavior of route discovery when using "naive" jitter (simply, delaying RREQ retransmission by a small uniformly distributed random delay), in order to identify and analyze the problems hereof, mostly related to route sub-optimality and excessive control traffic overhead. A *Window Jitter* mechanism is then proposed to address these issues – with the performance hereof, when compared to "naive" jitter being evaluated by way of modeling, theoretical analysis and experiments. The paper shows that the use of *Window Jitter* improves indeed the efficiency of route discovery in AODV and overcome the drawbacks identified for "naive" jitter.**

## I. INTRODUCTION

Since the late nineties, the Internet Engineering Task Force has been investing substantial efforts on the research, design, development and standardization of routing protocols for networks operating in increasingly challenging conditions, communication through more fragile and low-capacity links, with less pre-determined connectivity properties and with increasingly constrained router resources.

*1) MANET Protocol Developments:* The MANET working group converged on development of two protocol families: reactive protocols, including AODV [1], and proactive protocols, including the Optimized Link State Routing (OLSR) protocol [2]. A distance vector protocol, AODV operates in an *on-demand* fashion, acquiring and maintaining routes only while needed for carrying data, by way of a *Route Request-Route Reply* exchange. A link state protocol, OLSR is based on periodic control messages exchanges, each router proactively maintaining a routing table with entries for all destinations

in the network. A sizable body of work exists, including [3], studying the performance of these protocols in different scenarios, and justifying their complementarity [4]. For the purpose of this paper , it suffices to observe that OLSR, as a link state protocol, provides low delays and predictable, constant control overhead – at expense of requiring memory in each router for maintaining complete network topology. AODV limits the memory required for routing state to that for actively used routes – at the expense of delays for the *Route Request-Route Reply* exchange to take place, and control overhead dependent on data flows.

After acquiring operational experience, the MANET working group commenced developing successors to OLSR and AODV, denoted OLSRv2 and DYMO respectively. Whereas a relatively large and active community around OLSR standardized OLSRv2 [5] and [6], the momentum behind DYMO withered in the MANET working group. However, other derivatives of AODV have been implemented and used widely: IEEE 802.11s [7], for example, is based on AODV, and the G3-PLC standard [8], published in 2011, specifies the use of LOAD [9] (a simplified version of AODV) at the MAC layer, for providing mesh-under routing for utility (electricity) metering networks. Spurred by these experiences, 2011 saw the emergence of LOADng [10], as a successor to LOAD.

*2) Jitter:* In a wireless network, simultaneous packet transmissions by nearby nodes should be avoided because such might cause loss of transmissions on receivers due to collisions on the wireless channel. Depending on the characteristics of the medium access control (MAC) layer, in particular whether the retransmission of the unacknowledged transmissions is supported, such collisions may result in longer delays or even packet losses. A MAC protocol can, in some instances, handle this way of by collision detection and link-layer packet retransmissions, however, it is in other cases necessary to also have collision-avoidance mechanism in place at the network or higher layers also. In [11], jitter (adding a small, random delay on transmissions) is introduced on the interval between the periodic exchange of control messages. In RFC 5148 [5], the use of jitter is recommended for MANETs as one collision-avoidance mechanism for MANET routing protocol control traffic.

[12] introduces an analytical model for investigating the impact of the standardized jitter mechanism on network-wide

packet dissemination, and studied and quantified the additional delay incurred, the reduction in number of transmissions, and the effect of jitter in packet size. [13] presents the relationship between the maximal jitter duration and the probability of successful transmission, and different strategies of implementing jitter mechanisms are compared. [13] concludes that implementing jitter at any layer above IP (e.g. at the transport or application layer) brings virtually no benefits.

### A. Statement of Purpose

This paper studies the impact of jitter in route discovery in reactive routing protocols. Route discovery in reactive routing protocols uses flooding of Route Request (RREQ) messages through the network, in order to discover available routes from (requesting) sources to (requested) destinations. An intermediate node receiving an RREQ for the first time will process and, immediately, retransmit it. Using jitter permits reducing the probability that adjacent nodes will retransmit at the same time.

In addition to collision avoidance by way of a random delay on transmission of RREQ messages, this paper also considers:

- *Route discovery of optimal routes*. When RREQ messages are flooded through the network, the route cost (*e.g.,*hop count or any other link metrics) is also recorded. The destination of the RREQ will reply it with a Route Reply (RREP) message. However, the RREQ copy that arrives first may not always be the one which has traversed the most optimal path with respect to the metric used – and this is exacerbated by "naive" use of jitter.
- *Route discovery overhead*. In a simple flooding algorithm, duplicate messages are dropped by intermediate nodes, and not retransmitted. However, for RREQ flooding, in which the cumulated route cost is carried in the RREQ, intermediate nodes may need to transmit the same RREQ message multiple times – at least, when an RREQ arrives from the same source, to the same destination, and with the same sequence number as a previously forwarded RREQ, but with a lower route cost. Again, this is exacerbated by "naive" use of jitter.

The contribution of this paper is double. First, the impact of using jitter in route discovery in reactive routing protocols is studied, and problems arising from a too "naive" use of jitter are identified and quantified: additional overhead and sub-optimal paths. Second, a strategy denoted *Window Jitter*, is proposed, which retains the benefits of jitter while alleviating the issues identified from its naive application.

The remainder of this paper is organized as follows: section II provides a background by way of introducing the principal mechanisms of reactive routing protocols and of jitter. Section III analyzes the use of jitter for RREQ flooding in reactive routing protocols, and discusses the problems arising from the use of uniform jitter for this purpose. *Window Jitter* is proposed in section IV, followed by performance studies and comparisons in section V. The paper is concluded in section VI.

## II. REACTIVE PROTOCOLS AND APPLICATION OF JITTER

This section presents, in section II-A, a summary of the main mechanisms of reactive routing protocols, and describes, in section II-B, the basics principles of jitter applied for flooding, according to [5].

### A. Reactive Routing Protocol Principles and Overview

AODV [1] and its derivatives (LOAD [9], LOADng [10]) specify two principal operations: *Route Discovery* and *Route Maintenance*.

*1) Route Discovery:* During *Route Discovery*, RREQ messages are flooded through the network until they reach the sought destination – at which point that destination generates an RREP, which is unicast along the reverse path to the source of that RREQ. In AODV [1], intermediate routers can, furthermore, generate an RREP as reply to a received RREQ, provided that they have a "sufficiently recent" and valid route to the sought destination already. Additionally, an intermediate router generating an RREP also sends a *gratuitous RREP* to the sought destination, thus installing the complete forward and reverse paths. Protocols such as LOAD and LOADng do away with the intermediate/gratuitous RREPs, so that only the destination of the RREQ can reply, in order to simplify protocol operation and reduce the size of control messages.

*2) Route Maintenance:* *Route Maintenance* is performed when an actively used route fails, i.e., when a data packet cannot be delivered to the next hop towards the intended destination. When detecting that a route has failed, a Route Error (RERR) message is generated to notify the route failure.

### B. Jitter Technique for Flooding

Due to the shared nature of a wireless medium, simultaneous packet transmissions are likely to cause packet losses due to collisions. In order to prevent or minimize these collisions, RFC 5148 [5] recommends the use of jitter for different cases in which packets may be expected to be sent concurrently. One of the main examples of concurrent transmissions is packet flooding, *i.e.*, the procedure by which a packet is sent to all nodes in a network by way of having each node retransmit that packet exactly once on first receipt.

Without jitter, a node receiving a packet to be forwarded retransmits it immediately after processing. Instead, RFC 5148 [5] recommends that each node adds a small, random delay before forwarding a flooded packet, in order to avoid collisions with other , neighbouring nodes that may have received the same packet which might forward it over the same wireless channel as part of the flooding procedure. The recommendation from RFC 5148 [5] is that delays be selected following an uniform distribution between $0$ and a maximum jitter value, $J_m$. Figure 1 illustrates this use of jitter for flooding.

Other than prevention of packet collisions from simultaneous transmissions (from $B$ and $C$, in figure 1), the use of jitter in flooding has two immediate additional effects: (i) the flooding process is slowed, and (ii) nodes need larger buffers to store packets which have been received, but not yet forwarded. The tradeoff between these drawbacks and the reduction in the
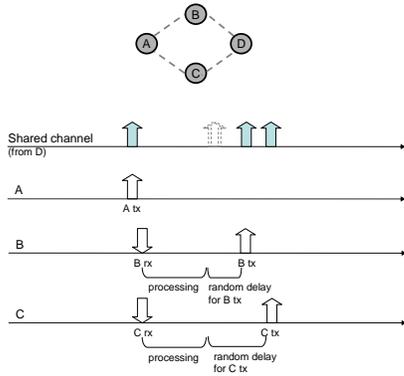
Figure 1. Use of jitter for flooding. Node *A* transmits a broadcast packet. Node *B* and *C* receive the packet at the same time and have roughly the same processing time. A collision will occur without jitter (the shadow in the shared channel). With jitter, the probability of a packet collision can be substantially reduced.

probability of collisions can be controlled by way of the length of the jitter interval, $J_m$.

## III. USE OF JITTER FOR ROUTE DISCOVERY IN REACTIVE ROUTING PROTOCOLS

This section discusses the use of jitter, as specified in RFC 5148 [5], for flooding of RREQ messages from a reactive protocol during route discovery. Section III-A describes, by way of an example, some drawbacks, incurred by the use of jitter for RREQ flooding, in terms of routing performance. This is related to the "*longer paths get first*" phenomenon, which is presented and described. Section III-B provides a probabilistic analysis of this phenomenon, which is further addressed in the remainder of this paper.

### A. Intuition

In a reactive routing protocol, RREQ messages from a source are flooded through the network in order to identify a route towards the intended destination. Deliberate jittering of RREQ retransmissions, in the way specified in RFC 5148 [5], is recommended by *e.g.,*LOADng [10]. However, the fact that RREQ messages reach their destination with a random delay has some drawbacks, in terms of path sub-optimality and/or control traffic inefficiency.
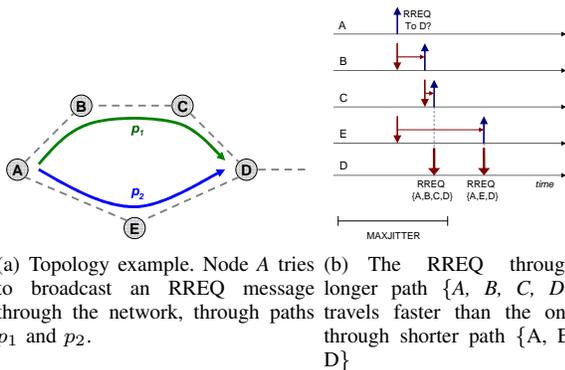


(a) Topology example. Node *A* tries to broadcast an RREQ message through the network, through paths $p_1$ and $p_2$.

(b) The RREQ through longer path {*A, B, C, D*} travels faster than the one through shorter path {A, E, D}

Figure 2. An example of delay inversion

Consider the topology shown in Figure 2(a), and assume that node $A$ floods (broadcasts) an RREQ to identify a route towards $D$. Under normal operation of a reactive routing protocol, *i.e.* without jitter, the RREQ would reach $D$ through the path $p_2 = \{A, E, D\}$ faster than through the path $p_1 = \{A, B, C, D\}$, assuming that processing times and transmission times at each intermediate node($T_i$), are similar. In fact, because the jitter $T_j$ is generally up to tens milliseconds, it is much greater than the processing time and transmission time (generally less than 1 millisecond, i.e., $T_j >> T_i$) in networks without significant congestion. For simplicity, only the delay caused by jitter is considered in the following analysis. The simulation results in V confirmed this assumption.

If a uniform random distribution $[0, J_m]$ is used at each hop, to determine an additional delay before retransmission, the message copy sent through the longer path (in number of hops), $p_1$, may reach the destination faster than the message copy over $p_2$ with a non-negligible probability, for instance if assigned jitter values are those shown in Figure 2(b).

In this case, node $D$ would reply to the Route Request from $A$ with a Route Reply (RREP) that advertises path $p_1$, which is suboptimal. When the RREQ traversing $p_2$ would reach $D$, $D$ would reply again to $A$'s Route Request with the (shorter) path $p_2$. This implies that $A$ would get, and possibly use for a certain amount of time, a suboptimal path towards $D$ ($p_1$), and it would need two RREP from $D$ in order to learn the optimal path from $A$ to $D$. If $D$ was not the destination of the requested route, but only an intermediate node towards that destination, then $D$ would retransmit twice the RREQ as it is received from $p_2$ and $p_1$.

This example illustrates that the use of uniform random distribution for jitter values when forwarding RREQ messages during route discovery in a reactive routing protocol may lead to cases in which "*transmissions over longer paths get first*" – this effect is hereafter denominated *delay inversion* caused by jitter, as it turns longer (worse) paths into paths which are traversed faster. These delay inversions are harmful due to at least three undesirable effects: (i) increase of sub-optimality of reported routes, (ii) growth of unnecessary RREQ flooded traffic, and (iii) growth of unnecessary RREP (unicast) traffic.

### B. Analysis

This section provides a quantitative probabilistic analysis of the delay inversion effect. Let $T_j$ be the random variable for jitter values, then $T_j \sim Uniform[0, J_m]$ according to RFC 5148 [5]. The delay caused by *uniform* jitter in an RREQ message traversing a path of $n$ hops, $T_1^{(n)}$, can be then be computed as follows:

$$T_1^{(n)} = \sum_{i=1}^{n} T_{j\,i} \qquad (1)$$

Given two paths between a source $X$ and a destination $Y$, with lengths $n$ and $m$, let $D_1^{(n,m)}$ be the *inter-path delay difference*, *i.e.*, the difference between jitter delays suffered by an RREQ flooded through two paths between $X$ and $Y$,

of $n$ and $m$ hops. It is a random variable that depends on the random variables for the jitter values in the way shown in (2).

$$D_1^{(n,m)} = T_1^{(n)} - T_1^{(m)} = \sum_{i=1}^{n} T_{j_i} - \sum_{j=1}^{m} T_{j_j} \qquad (2)$$

The probability of inversion, that is, the probability that the delay inversion ("*longer paths get first*" effect) occurs in the RREQ flooding, corresponds to $P(D_1^{(n,m)} > 0|_{n<m})$, whose expression is detailed in (3).

$$P\left(T_1^{(m)} < T_1^{(n)}\right) = P\left(D_1^{(n,m)} > 0\right) = \int_0^\infty f_{D_1^{(n,m)}}(t)dt \qquad (3)$$

The probability density function (PDF) of $D_1^{(n,m)}$, $f_{D_1^{(n,m)}}(t)$, has the following expression:

$$f_{D_1^{(n,m)}}(t) = \left(\bigotimes_{i=1}^{n+m} f_{T_{j_i}}\right)(t + mJ_m) \qquad (4)$$

where $\otimes$ denotes the convolution. This expression can be computed by using the well-known result for the PDF of the sum, $S_n$, of $n$ independent uniform random variables $Uniform[0,1]$ [14]:

$$f_{S_n}(x) = \frac{1}{(n-1)!} \sum_{k=0}^{n} (-1)^k \binom{n}{k} (x-k)_+^{n-1} \qquad (5)$$

Note that $f_{S_n}(x) = 0$ for $x > n$ and $x < 0$. Notation $z_+$ denotes the positive part of $z$, *i.e.*, $z_+ = z$ if $z > 0$, $z_+ = 0$ otherwise. Then, the probability of delay inversion between two paths with lengths $n$ and $m$ ($n < m$), becomes $P(D_1^{(n,m)} > 0)|_{n<m}$. The expression of $P_1 \equiv P(D_1^{(n,m)} > 0)$ is computed in (6):

$$P_1 = \frac{1}{(n+m)!} \sum_{k=0}^{n+m} (-1)^k \binom{n+m}{k} \left((n+2m-k)^{n+m} - (m-k)_+^{n+m}\right) \qquad (6)$$

Figure 3(a) illustrates the theoretical values for the probability of inversion for different values of $n$ and $m$, *i.e.*, the probability that a path of $m$ hops performs faster forwarding than a path of length $n$. Figure 3(b) displays the same probability for different values of path length $m$, for cases in which $n < m$. Both the theoretical values and the results from a discrete-event simulation (each point corresponding to the averaged value over 200 samples) are displayed. Note that both Figures 3(b) shows bidimensional cuts of the surface presented in Figure 3(a) – these cuts result from the intersection of this surface with planes $\pi_1 : \{m = ct.\}$ and $\pi_2 : \{\Delta = m - n = ct.\}$, respectively.

Expression (6) indicates that the delay inversion occurs, under the conditions specified in RFC 5148 [5], with a significant probability.
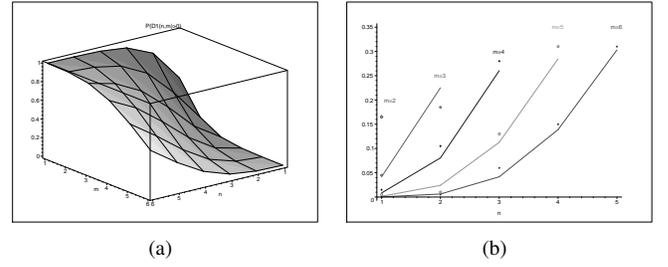


Figure 3. **(a)** Theoretical value of $P_1 \equiv P(D_1^{(n,m)} > 0)$, for $1 \le n \le 6$, $1 \le m \le 6$; **(b)** Restriction of $P_1$ for $m = 1, 2, 3, 4, 5$, $n < m$ (theoretical values in lines, simulations in points).

Two aspects can be highlighted from this analysis: (i) from (6), the probability of inversion does not depend on the length of the jitter interval, $J_m$, meaning that it cannot be addressed by modifying the jitter interval length; and (ii) from Figures 3(b) , probability of inversion does not only depend on the difference between path lengths, $\Delta = m - n$, but also on the absolute values of path lengths $n$ and $m$: as paths become longer, more random jitter values are assigned to an RREQ message and it is more likely that delay inversions occur.

## IV. THE WINDOW JITTER

This section presents a mechanism for substantially reducing the number of situations in which RREQ messages from a source reach the destination faster through a longer (and sub-optimal) path rather than through a shorter path. As this effect cannot be addressed by way of the $J_m$ parameter, this section proposes to modify the distribution of jitter values – the *window jitter* distribution.

### A. Rationale and Definition

With the uniform distribution for jitter values specified in RFC 5148 [5], the longer the path that a flooded RREQ traverses, the longer is the interval of possible delay values $T_1^{(n)}$ for that flooded RREQ. From (1), the upper bound for total jitter delay grows linearly with the length of the path, but the lower bound stays fixed at zero. This is corrected in the modified distribution by introducing a minimum jitter interval in each hop. Jitter values are then instances of a random variable $T_j^* \sim Uniform[\alpha J_m, J_m]$, where $\alpha \in (0,1)$ and $\alpha J_m$ is a minimum jitter value. Note that $\alpha = 0$ corresponds to the jitter distribution specified in RFC 5148 [5], $\alpha = 1$ would imply a deterministic delay (of length $J_m$), and $\alpha \ne 0$ implies that the lower bound for the RREQ delay grows linearly with the length of the traversed path.

Figure 4 shows the probability density functions (PDFs) for the jitter value as specified in RFC 5148 [5] ($T_j$) and the window jitter random variable ($T_j^*$).

The most obvious consequence of using the random variable $T_j^*$ instead of the random variable $T_j$ for assigning jitter values is, that the average value of jitter increases from $\mathbb{E}\{T_j\} = \frac{1}{2}J_m$ to $\mathbb{E}\{T_j^*\} = \frac{1+\alpha}{2}J_m$. Upper bound (worst case) for the delay caused by jitter, however, is $J_m$ in both cases.
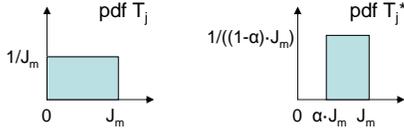
Figure 4. PDFs of random variables $T_j$ based on RFC 5148 [5] and $T_j^*$ based on window jitter.

### B. Analysis

The analysis performed in section III-B for $T_j$ can be repeated for $T_j^*$, and that brings the following expressions for the total jitter delay, $T_2^{(n)}$:

$$T_2^{(n)} = \sum_{i=1}^{n} T_j^*{}_i \tag{7}$$

And the inter-path delay difference for the window jitter distribution, $D_2^{(n,m)}$:

$$D_2^{(n,m)} = T_2^{(n)} - T_2^{(m)} = \sum_{i=1}^{n} T_j^*{}_i + \sum_{j=1}^{m} \left(-T_j^*{}_j\right) \tag{8}$$

It is immediate to observe that variable $D_2^{(n,m)}$ has the following PDF:

$$f_{D_2^{(n,m)}}(t) = \left(\bigotimes_{i=1}^{n+m} f_{T_j^*{}_i}\right)(t + mJ_m) \tag{9}$$

Without loss of generality, and for the sake of simplicity, it can be assumed in the following that $J_m = 1$. Then, also using (5), the PDF detailed in (9) can be expressed in the terms of *Proposition 1*:

**Proposition 1.**

$$f_{D_2^{(n,m)}}(t) = \frac{1}{1-\alpha} \frac{1}{(n+m-1)!} \times$$
$$\times \sum_{k=0}^{n+m} (-1)^k \binom{n+m}{k} \left(\frac{t-n\alpha+m}{1-\alpha} - k\right)_+^{n+m-1} \tag{10}$$

*for* $\frac{t-\alpha n+m}{1-\alpha} \in [0, n+m]$, *and* 0 *otherwise.*

Therefore, the probability of delay inversion with paths of length $n$ and $m$, $P(D_2^{(n,m)} > 0)|_{n<m}$, can be computed as indicated in *Proposition 2*:

**Proposition 2.** *The expression of* $P_2 \equiv P(D_2^{(n,m)} > 0)$ *corresponds to:*

$$P_2 = \frac{1}{(n+m)!} \left(\sum_{k=0}^{n+m} (-1)^k \binom{n+m}{k} (n+m-k)^{n+m} - \right.$$
$$\left. - \sum_{k=0}^{k^*} (-1)^k \binom{n+m}{k} \left(\frac{m-n\alpha}{1-\alpha} - k\right)^{n+m}\right) \tag{11}$$

*where* $k^* = \lfloor \frac{m-n\alpha}{1-\alpha} \rfloor_+$, *for* $\alpha \leq \frac{n}{m}$. $P_2 = 0$ *otherwise (i.e., for* $\alpha > \frac{n}{m}$).

*Proof:* From the definition of probability, $P(D_2 > 0) = \int_0^\infty f_{D_2}(x)dx$, which is:

$$P_2 = \int_0^\infty \frac{dt \sum_{k=0}^{n+m} (-1)^k \binom{n+m}{k} \left(\frac{t-n\alpha+m}{1-\alpha} - k\right)_+^{n+m-1}}{(1-\alpha)(n+m-1)!} =$$
$$= \frac{\sum_{k=0}^{n+m} (-1)^k \binom{n+m}{k} \int_0^\infty dt \left(\frac{t-n\alpha+m}{1-\alpha} - k\right)_+^{n+m-1}}{(1-\alpha)(n+m-1)!} \tag{12}$$

The integrand of (12) is zero when $\frac{t-n\alpha+m}{1-\alpha} > n+m \iff t > n - m\alpha$, and $n - m\alpha > 0$ (see *Proposition 1*), therefore (12) can be rewritten as follows:

$$P_2 = \frac{\sum_{k=0}^{n+m} (-1)^k \binom{n+m}{k} \int_0^M dt \left(\frac{t-n\alpha+m}{1-\alpha} - k\right)_+^{n+m-1}}{(1-\alpha)(n+m-1)!} \tag{13}$$

where $M \equiv max\{0, n - m\alpha\}$. Consider the integral $I_0 \equiv \int_0^M dt \left(\frac{t-n\alpha+m}{1-\alpha} - k\right)_+^{n+m-1}$. The integrand of $I_0$ is non-zero when:

$$\frac{t - n\alpha + m}{1-\alpha} - k > 0 \implies t > k(1-\alpha) + n\alpha - m \equiv t_0 \tag{14}$$

and zero otherwise. Therefore, integration limits of $I_0$ can be modified as follows:

$$I_0 = \int_{max\{0,t_0\}}^{max\{0,n-m\alpha\}} dt \left(\frac{t - n\alpha + m}{1-\alpha} - k\right)^{n+m-1} \tag{15}$$

$t_0$ depends on $k$. It is worth to observe that:
- $t_0 < n - m\alpha, \forall k : 0 \leq k \leq n+m$. And $n - m\alpha > 0$ iif $\alpha > \frac{n}{m}$.
- $t_0 > 0 \iff k(1-\alpha) + n\alpha - m > 0$, for $k_{min} = 0$ and $k_{max} = n+m$. This implies that $\alpha \leq \frac{n}{m}$ and $\alpha > \frac{m}{n}$.
- For $\alpha \leq \frac{n}{m}$, $\alpha \leq \frac{m}{n}$, there is a $k^* = \lfloor \frac{m-n\alpha}{1-\alpha} \rfloor_+ \geq 0$ such that $t_0 > 0, \forall k > k^*$.

Which implies that $I_0$ can be expressed as:

$$I_0 = \int_{max\{0,t_0\}}^{max\{0,n-m\alpha\}} dt \left(\frac{t - n\alpha + m}{1-\alpha} - k\right)^{n+m-1} \tag{16}$$

Then, (13) can be computed to Proposition 2, while $\alpha \leq \frac{n}{m}, \frac{m}{n}$

$\blacksquare$

Figures 5(a) and 5(b) show the value of $D_2^{(n,m)}$ for different combinations of path lengths $n$ and $m$. In the analysis, hop-count metric is considered, i.e., the routes with less hops are more preferred. $\alpha$ is set to $\frac{1}{2}$, to have a balance between the randomness of jitter, and the "width" of window to reduce *delay inversion* effect.

It can be observed in figure 5.a that transition from values close to 0 to 1 (*i.e.*, from situations in which RREQ transmissions over the $n$-path are never faster than those over the $m$-path, to situation in which are always faster) is significantly

more steep with the modified (generalized) distribution of jitter values than with the distribution of RFC 5148 [5] (see figure 3(a)). As the ideal situation would be that $D_{\text{ideal}}^{(n,m)} = 1$ for $n > m$ and $D_{\text{ideal}}^{(n,m)} = 0$ for $n < m$, the use of the modified distribution makes the jitter performance closer to that ideal behavior.
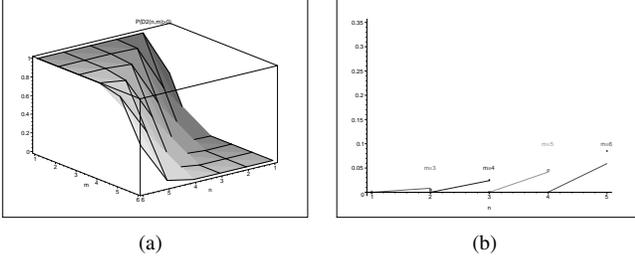


(a)                              (b)

Figure 5. For $\alpha = \frac{1}{2}$, **(a)** Theoretical value of $P_2 \equiv P(D_2^{(n,m)} > 0)$, for $1 \leq n \leq 6$, $1 \leq m \leq 6$; **(b)** Restriction of $P_2$ for $m = 1, 2, 3, 4, 5$, $n < m$ (theoretical values in lines, simulations in points).

Figure 5(b) shows the probability of delay inversion for the modified distribution of jitter values, depending on the difference $\Delta = m - n$, for different values of $n$ and $m$. As in figure 3(b), theoretical values (lines) and simulations (points, each of them averaged over 200 samples) are displayed together. It can be observed that the values are substantially lower than those achieved with $T_j \sim Uniform[0, J_m]$: for very similar ($\Delta = m - n = 1$, which is the most frequent case) and long paths ($n = 5$), the probability reduces in a factor 5 and stays below the 6%; the relative variation becomes still more significant as paths are shorter.

### C. Generalization: Another Perspective

The uniform distribution between $0$ and $J_m$ specified in RFC 5148 [5] is the maximum entropy distribution among those assigning continuous jitter values between $0$ and $J_m$ [15]. This implies that the use of this distribution maximizes the randomness of the total delay suffered by an RREQ message sent along a certain path.

The window jitter described in this section reduces randomness and introduces a (deterministic) dependency of the total RREQ delay to the length $n$ of the traversed path. When assigning jitter values according to the distribution of random variable $T_j^*$, the total delay caused by jitter in a path of $n$ hops belongs to the interval $[n\alpha J_m, n J_m]$ ($\alpha \neq 0$). The trade-off between randomness and path length deterministic dependence can be controlled by way of parameter $\alpha \in (0, 1)$: the closer $\alpha$ is to 1, the more deterministic becomes the total delay of an RREQ message with respect to the path length.

Each additional hop in the path traversed by an RREQ message causes at least an additional delay of $\alpha J_m$ before being delivered to the requested destination. This is done in order to increase the probability that the RREQ message traverses first a shorter path, in number of hops, rather than a longer path, considered worse for routing. This model thus assumes that shorter paths are preferable to longer paths, that is, a hop count metric is implicitly assumed.

The window jitter principle can, naturally, be extended to non-trivial link metrics, *e.g.*, based on probability of successful transmission (ETX [16]). Given a link quality indicator $LQ \in (0, 1)$ ($LQ \longrightarrow 1$ for high quality links), jitter values would be assigned under a generalized window jitter distribution uniformly within the interval $[(1 - LQ)J_m, J_m]$, in order to reduce the probability of delay inversion or, equivalently, increase the probability that an RREQ message is forwarded faster by routers receiving it on better links. Note that the window jitter distribution presented and analyzed in sections IV-A and IV-B corresponds to the particular case of $LQ = 1 - \alpha$ for all available links.

### V. NETWORK SIMULATION AND ANALYSIS

The window jitter distribution proposed in section IV is implemented, studied and evaluated by way of network simulations over different scenarios, under several different configurations. These simulations permit validating the theoretical results obtained in sections III and IV, comparing in practice the performance of AODV RREQ flooding when using the original distribution of RFC 5148 and the proposed window jitter distribution. Simulations also allow to identify the networking and jitter elements that have impact in this performance. Section V-A presents the setting of the performed network simulations. Section V-B describes the main results obtained in the experiments.

### A. Simulation Setting

In order to understand the impact in flooding performance of different jitter settings, ns2-simulation results are presented in the below. Simulations were made of a field of $1000 \times 1000$ meters, with varying numbers of routers placed randomly, equipped with 802.11b radio interface and with 250m range. For the purpose of this study, router mobility was not considered. The metric is based on hop-count.

Each simulation lasts for 100 seconds. Thirty random routers in the network initiate route discovery to another random destination every two seconds. The number of collisions, average overhead, average route discovery delay and average path length are measured.

Different jitter settings are compared:
- *No jitter*.
- *Standard RFC 5148 jitter, $J_m = 100ms$.* Jitter is selected within $[0, 100]$ ms (mean, $50ms$).
- *Standard RFC 5148 jitter, $J_m = 200ms$.* Jitter is selected within $[0, 200]$ ms (mean, $100ms$).
- *Window jitter, $\alpha = \frac{1}{2}$, $J_m = 100ms$.* Jitter is selected within $[50, 100]$ ms (mean, $75ms$).
- *Window jitter, $\alpha = \frac{2}{3}$, $J_m = 150ms$.* Jitter is selected within $[100, 150]$ ms (mean, $125ms$).

For each of the five settings, two different strategies of RREQ forwarding scheme are considered:
- *Shortest-path RREQ forwarding*. The routers attempt to build the shortest path to the destination, *i.e.*, the intermediate routers will forward the RREQ with shorter path, even if this RREQ has been already processed before.

This approach is used in most of the implementations like [10].

- *Shortest-delay RREQ forwarding*. The routers only forward the RREQ that arrives first. All the following RREQ messages are ignored, even if it traveled through a better path. This might result in non-optimal paths.

### B. Simulation Results

*1) Shortest-path RREQ forwarding:* The first observation that can be from the shortest-path RREQ forwarding results is that the use of standard jitter, with the distribution specified in RFC 5148, does not have an outstanding impact on the RREQ flooding performance, when compared with the *no-jitter* setting – neither in terms of collisions, control overhead or data path length. Differences can be observed, in contrast, between *window jitter* settings and the other settings.

Figure 6(a) shows the number of collision with different density of nodes (routers). The setting with *no jitter* has the highest number of collisions, slightly higher than the *standard jitter* settings, because adjacent nodes are more likely to retransmit received RREQs at the same time. The *window jitter* (50-100 ms, 100-150 ms) settings have significantly less collisions, especially in high-density scenarios. This is because the use of the window jitter setting enables forwarding routers to reduce the number of transmissions (*i.e.*, overhead) by reducing the cases of delay inversion, as shown in Figure 6(b).



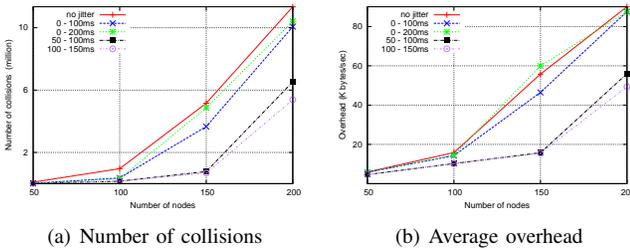(a) Number of collisions   (b) Average overhead

Figure 6. Simulation results of shortest-path RREQ forwarding

Concerning the evolution of route discovery delay (*i.e.*, the time between RREQ transmission and reception of the corresponding RREP) with respect to the network density in the shortest-path RREQ forwarding strategy (Figure 7(b)), it is interesting to observe that the impact of the evaluated jitter settings is different for low-density and high-density networks. In the first case, the average route discovery delay for each jitter setting is strongly related to the mean of the jitter random variable: settings with higher jitter means present higher average delays in the route discovery processes. For denser networks, in contrast, window jitter distributions present a better performance in terms of route delivery delay than standard jitter distributions, regardless of the mean value of the jitter random variable. The use of standard jitter distributions (or the immediate retransmission of RREQ messages, without jitter) in dense networks leads to an explosion of control traffic when the shortest-path RREQ forwarding strategy is used. This control traffic explosion can be indirectly observed in the evolution of the number of packet collisions in the network

(Figure 6(a)) and the data packet delivery ratio (Figure 7(a)) – as the control traffic load grows beyond the network capacity, a more significant fraction of transmitted data packets cannot be correctly delivered even when routes are available, due to the increasing number of packet collisions.

By reducing substantially the probability of delay inversion, the use of window jitter distributions improves the quality of the selected routes (see Figure 7(c)) and allows to reduce the number of Route Replies sent in response to a Route Request. This alleviates the control traffic load of the network and decreases the number of packet collisions, therefore reducing significantly the average delay for route discovery processes.

*2) Shortest-delay RREQ forwarding:* As a comparison study, the shortest-delay RREQ forwarding approach is also simulated. Because only the first RREQ is forwarded, the network tends to have the same overhead, therefore the number of collisions is similar in different jitter settings, as shown in Figure 8(a) – the number of collisions in no-jitter settings, in contrast, is obviously higher. In this situation of similar control traffic overhead, those settings using *windows jitter* distributions have longer average route discovery delay, as expected (Figure 8(b)) – they have also higher mean values for the jitter values.

However, because the intermediate nodes simply forward the RREQ that arrives first, and all the following RREQs are ignored, the *no jitter* and *normal jitter* settings got sub-optimal routes. In the meantime, with *window jitter*, a much shorter routes can be explored, as illustrated in Figure 8(c). This is more interesting for less time-critical but power-constrained networks (such as sensor networks).

### VI. CONCLUSION

The use of jitter for flooding has proved beneficial for wireless networks: by adding a random delay before forwarding a broadcast message, the number of collisions can be substantially reduced. Proactive routing protocols, such as OLSR, has found the use of an uniformly distributed jitter on flooded link-state messages to dramatically improve performance, and this has therefore been recommended in RFC 5148 [5].

Alas, in case of route discovery in reactive routing protocols, this recommendation is less beneficial: using an uniform distribution of jitter for when flooding RREQ messages for the delays has some very undesirable effects, including that, with non-negligible probability, the route discovery process of such reactive protocols may result suboptimal routes towards requested destinations. This paper has identified and quantified this effect, called delay inversion.

In order to reduce the probability and impact of delay inversion, this paper proposes a modified jitter distribution – window jitter: in its most general version, it consists of taking the quality of traversed links into account in the computation of jitter delays before retransmission of RREQ by intermediate nodes. This, in a way such that messages are forwarded faster over better links, and therefore a higher probability of the route discovery process resulting optimal routes. Theoretical
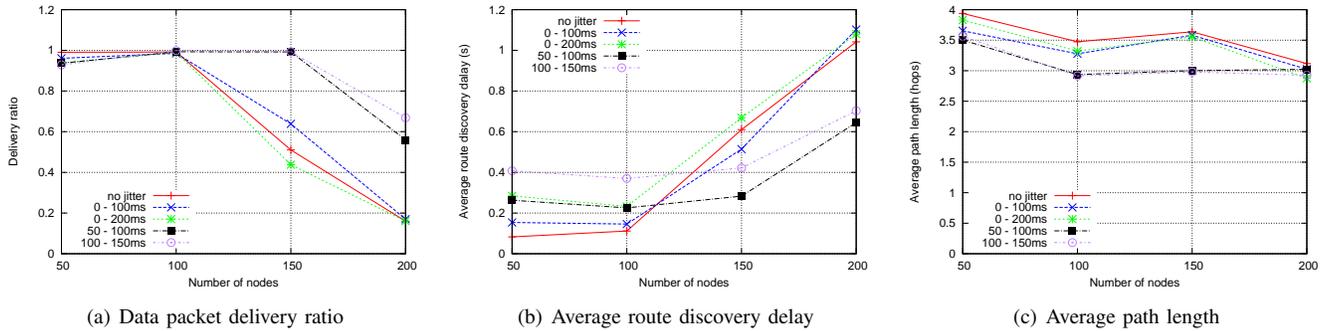
(a) Data packet delivery ratio     (b) Average route discovery delay     (c) Average path length

Figure 7.   Simulation results of shortest-path RREQ forwarding



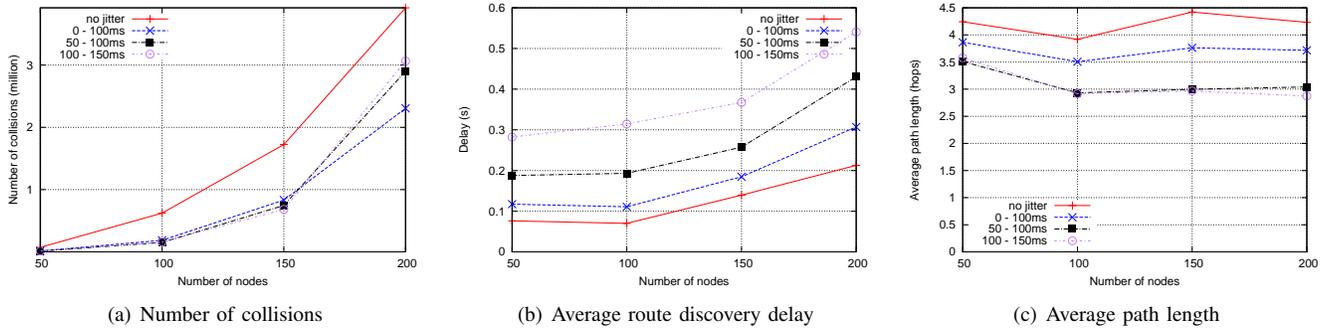(a) Number of collisions     (b) Average route discovery delay     (c) Average path length

Figure 8.   Simulation results of shortest-path RREQ forwarding

analysis of window jitter (with hop count metrics, although the results generalise to also other metrics) shows that, despite its simplicity, this modification is able to, very substantially, reduce the probability of experiencing the effect of delay inversion, manifested by sub-optimal routes resulting from the route discovery process, with the cost of slightly increasing the route discovery delay. The benefits of the window jitter distribution, as compared to the uniform jitter distribution recommended in RFC 5148 [5], are also confirmed by way of simulations.

## REFERENCES

[1] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," Experimental RFC 3561, July 2003.

[2] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)," Experimental RFC 3626, October 2003.

[3] T. Clausen, P. Jacquet, and L. Viennot, "Comparative study of routing protocols for mobile ad-hoc networks." Proceedings of the IFIP MedHocNet, September, Sardinia, Italy, 2002.

[4] T.Clausen, P.Jacquet, and L.Viennot, "Analyzing control traffic overhead versus mobility and data traffic activity in mobile ad hoc network protocols," *ACM Journal on Wireless Networks, vol. 10 no. 4*, 2004.

[5] T. Clausen, C. Dearlove, and B. Adamson, "Jitter Considerations in MANETs," IETF Inf. RFC 5148, February 2008.

[6] T. Clausen, C. Dearlove, and P. Jacquet, "The Optimized Link State Routing Protocol version 2," Internet Draft, draft-ietf-manet-olsrv2-11, work in progress, April 2010.

[7] G. Hiertz, S. Max, R. Zhao, D. Denteneer, and L. Berlemann, "Principles of ieee 802.11s," in *Proceedings of WiMAN in conjunction with the 16th ICCCN*, Honolulu, Hawaii, USA, Aug 2007, p. 6.

[8] "ITU-T G.9956: Narrow-Band OFDM power line communication transceivers - Data link layer specification," November 2011.

[9] K. Kim, S. D. Park, G. Montenegro, S. Yoo, and N. Kushalnagar, "6LoWPAN Ad Hoc On-Demand Distance Vector Routing," June 2007, Internet Draft, work in progress, draft-daniel-6lowpan-load-adhoc-routing-03.

[10] T. Clausen, A. C. de Verdiere, J. Yi, A. Niktash, Y. Igarashi, H. Satoh, and U. Herberg, "The lln on-demand ad hoc distance-vector routing protocol - next generation," The Internet Engineering Task Force, October 2011, internet Draft, work in progress, draft-clausen-lln-loadng.

[11] T.Clausen, G.Hansen, L.Christensen, and G.Behrmann, "The optimized link state routing protocol, evaluation through experiments and simulation." Proceedings of the IEEE conference on Wireless Personal Multimedia Communications (WPMC), 2001.

[12] J. A. Cordero, P. Jacquet, and E. Baccelli, "Impact of jitter-based techniques on flooding over wireless ad hoc networks: Model and analysis." Proceedings of the IEEE INFOCOM 2012, Orlando, USA, 2012.

[13] R. Friedman, D. Hay, and G. Kliot, "Jittering Broadcast Transmissions in MANETs: Quantification and Implementation Strategies," Department of Computer Science, Technion, Tech. Rep., 2009. [Online]. Available: http://www.cs.technion.ac.il/ gabik/publications/Jitter-submit2009.pdf

[14] W. Feller, *An Introduction to Probability Theory and its Applications, Vol. 1*, 3rd ed. John Wiley & Sons, 1950.

[15] Karmeshu, *Entropy Measures, Maximum Entropy Principle and Emerging Applications*. Springer, 2003.

[16] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (MobiCom)*, San Diego, California, USA, Sep 2003, p. 8.