An Adaptive Jitter Mechanism for Reactive Route Discovery in Sensor Networks

Juan Cordero, Jiazi Yi, Thomas Heide Clausen

▶ To cite this version:

HAL Id: hal-02263370

https://hal-polytechnique.archives-ouvertes.fr/hal-02263370

Submitted on 4 Aug 2019

Article

An Adaptive Jitter Mechanism for Reactive Route Discovery in Sensor Networks

Juan Antonio Cordero [1], Jiazi Yi [2,*] and Thomas Clausen [2]

[1] Institute of Information and Communication Technologies, Electronics and Applied Mathematics (ICTEAM), Université catholique de Louvain, Louvain-la-Neuve B-1348, Belgium;
E-Mail: j.a.cordero@gmail.com

[2] Laboratoire d'Informatique (LIX), École Polytechnique, Palaiseau 91120, France;
E-Mail: thomas@thomasclausen.org

* Author to whom correspondence should be addressed; E-Mail: jiazi@jiaziyi.com;
Tel.: +33-1-77-57-80-85.

Abstract: This paper analyses the impact of jitter when applied to route discovery in reactive (on-demand) routing protocols. In multi-hop non-synchronized wireless networks, jitter—a small, random variation in the timing of message emission—is commonly employed, as a means to avoid collisions of simultaneous transmissions by adjacent routers over the same channel. In a reactive routing protocol for sensor and *ad hoc* networks, jitter is recommended during the route discovery process, specifically, during the network-wide flooding of route request messages, in order to avoid collisions. Commonly, a simple uniform jitter is recommended. Alas, this is not without drawbacks: when applying uniform jitter to the route discovery process, an effect called delay inversion is observed. This paper, first, studies and quantifies this delay inversion effect. Second, this paper proposes an adaptive jitter mechanism, designed to alleviate the delay inversion effect and thereby to reduce the route discovery overhead and (ultimately) allow the routing protocol to find more optimal paths, as compared to uniform jitter. This paper presents both analytical and simulation studies, showing that the proposed adaptive jitter can effectively decrease the cost of route discovery and increase the path quality.

Keywords: jitter; reactive routing protocol; route discovery; overhead

1. Introduction

Flooding is a key networking operation in wireless multi-hop networks, used for the dissemination of data packets and able to operate in a dynamic network, even in the absence of (or not relying on) accurate information about the network topology. For wireless routing protocols (reactive, proactive or both), flooding is for acquiring or disseminating topological information. In its most basic form, flooding consists of requiring every router in the network to retransmit each received packet exactly once over all of its interfaces. This mechanism leads to the well-known broadcast storm problem [1], a consequence of collisions between concurrent and unnecessary retransmissions of the flooded packet. To counter this problem, more sophisticated mechanisms for efficient flooding have been proposed, implemented and standardized [2–4] to reduce the number of (redundant) retransmissions and, consequently, the amount of packet collisions occurring. Some of these proposals substantially improve the performance of flooding in wireless multi-hop networks.

However, the elimination of redundant retransmissions is not sufficient for completely avoiding packet collisions. Even if no redundant transmissions are present, wireless flooding in self-organized, decentralized networks leads to packet collisions caused by the simultaneous transmission of adjacent routers over the same wireless channel. These collisions constitute an important source of packet losses in wireless flooding.

Different approaches have been explored to address this issue and to minimize its impact in wireless multi-hop networks. Classic MAC (Medium Access Control) collision avoidance mechanisms [5,6] are not suited to current wireless sensor scenarios and are unable to solve all possible cases of collisions (e.g., broadcast or multicast transmissions, collisions between non-neighboring routers). Recent research efforts [7,8] have focused on other alternatives, such as the use of multi-channel assignments in wireless sensor networks. These approaches are able to reduce the problem of collisions in potentially dense networking scenarios, at the cost of adding an additional complexity layer (or relying on previous knowledge of the network topology) and renouncing the semi-broadcast capability of the wireless network.

According to the IETF (Internet Engineering Task Force), the problem of packet collisions in a MANET (Mobile Ad hoc Network) can be further alleviated by introducing jitter (a small, random delay on transmissions) in the network layer. In RFC 5148 [9], the use of jitter is recommended for MANETs and wireless sensor networks as a simple collision avoidance mechanism for routing protocol control traffic, such as periodically scheduled packets or event-triggered packets in the Optimized Link State Routing (OLSR) protocol [10,11] or for MANET-enabled areas in OSPF (Open Shortest Path First) routing [12,13].

1.1. Related Work

Subsequent standardization of jitter by the IETF in RFC 5148 [9], its implementation in different routing protocols and research to evaluate and discuss the impact of these techniques in the performance of the protocols making use of them have been undertaken. Friedman *et al.* [14] presented the relationship between the maximal jitter duration and the probability of successful transmission and provided a comparison between different strategies of implementing jitter mechanisms. This paper

concluded that implementing jitter at any layer above the IP (Internet Protocol) layer (e.g., at the transport or application layer) brings virtually no benefits. Cordero *et al.* [15] introduced an analytical model for investigating the impact of the standardized jitter mechanism for link-state flooding (which includes jittering and packet piggybacking; see RFC 5148 [9] for details) on network-wide packet dissemination performance. Based on that model, [15] studied and quantified the additional delay incurred, the reduction in the number of transmissions and the effect of jitter in packet size. While this paper focuses on flooding in reactive routing, it borrows model assumptions and terminology from [15] in order to study analytically the impact of jittering. Similarly, [16,17] discuss and present preliminary studies of the issues related to the application of jitter in reactive routing protocols, which are extended and completed in this paper.

This paper studies the use of jitter for improving the performance of reactive (on-demand) routing in wireless multi-hop sensor networks. Reactive routing protocols use flooding to disseminate route request messages in the network until they reach their destination, and the introduction of jitter techniques in route discovery is recommended to minimize systematic collisions. The "delay inversion" effect when using traditional uniform jitter [9] is identified and described, and an adaptive jitter mechanism is then proposed to reduce the route discovery overhead and improve path optimality. Both theoretical and experimental (simulation-based) analyses are performed and discussed; the corresponding results show that the proposed adaptive jittering substantially improves the flooding performance with little cost.

1.2. Outline

This paper studies the optimization of jitter mechanisms for route discovery of reactive routing protocols and is organized as follows: Section 2 presents the basics of reactive routing protocols, describes the jitter mechanism for flooding optimization described in RFC 5148 [9] and discusses the main drawbacks of the application of jitter in this form to reactive routing protocols. The paper then introduces and studies the "delay inversion" effect, a specific issue related to the use of standard jittering for route request flooding in reactive protocols. Section 3 describes this effect, proposes an alternative for the jitter distribution (the "adaptive jitter") and discusses some additional variations of this alternative. Section 4 compares analytically the behavior of the standard (uniform) jitter and the proposed window jitter with respect to the "delay inversion" effect and also examines analytically other side effects of jittering in flooding performance. This is complemented with an experimental performance comparison of the proposed techniques and standard jitter. The characteristics and algorithmic performance of the compared techniques and variants are studied in Section 5 by way of graph simulations. The performance impact of the different jittering techniques, when applied to reactive routing protocols, is studied in Section 6 by way of network simulations. Section 7 discusses the results presented throughout the paper and examines the performance trade-off illustrated in the comparison between the considered jitter techniques and variations. Finally, this paper is concluded in Section 8.

2. Reactive Routing Protocols and Application of Jitter

This section introduces the basic operations of reactive routing protocols, as used in some wireless *ad hoc* and sensor networks. Then, the use and impact of jitter on flooding performance are briefly introduced and discussed.

2.1. Basic Operations of Reactive Routing Protocols

In a reactive routing protocol, routes are computed on demand, *i.e.*, a router seeks to construct and maintain paths to a destination, only when it has data to deliver to that destination. Consequently, control traffic is generated in response to data traffic only; absent data traffic, no control traffic is generated. Furthermore, as long as data traffic flows along an already established path and no errors are incurred, no control traffic is generated for maintaining the path. Finally, each router is required to maintain the state only for the active paths, on which it is an intermediary, as opposed to, e.g., a complete topology map, as in a classic link state protocol. Unsurprisingly, the main mechanisms in a reactive routing protocol are denoted route discovery and route maintenance.

Reactive routing protocols have been explored in academic literature, as well as have been published as experimental protocols by the IETF: the Ad hoc On-Demand Distance Vector (AODV) protocol, as RFC 3561 [18] in 2003, and the Dynamic Source Routing (DSR) protocol, as RFC 4728 [19], in 2007. Reactive protocols typically find use in wireless *ad hoc* and sensor networks, due to the limited amount of states required in each router. For example, LOADng (Light-weight On-demand Ad hoc Distance-vector routing–Next Generation)—a simplified and extended version of AODV [20]—is the routing protocol component of the G3-PLC (Power Line Communication) ITU-T (International Telecommunication Union Telecommunication Standardization Sector) standard for communication in the "smart grid" [21].

2.1.1. Route Discovery

During route discovery, route request (RREQ) messages are flooded through the network, with each intermediary router forwarding the RREQ also recording the reverse path, *i.e.*, the next hop towards the originator of the RREQ. When the RREQ reaches the sought destination, that destination generates an RREP (route reply), which is unicast along the installed reverse path. RREQ and RREP messages carry a monotonically-increasing sequence number, permitting both duplicate detection and detecting which of the two messages contains the most "fresh" information. The RREQ message carries also the metric semantics of a path, *i.e.*, the "cost" of the path to the originator is recorded and updated by each intermediate router. Two flooding modes are possible: the shortest-delay mode and the shortest-path mode. Depending on the flooding mode, RREQ forwarding and RREP generating rules may be slightly different.

> **Shortest delay mode** Routers in the network only forward the first RREQ message received from a given source to a given destination; later arriving RREQs with the same pair (source, sequence_number) will be dropped, even if they advertise better paths than the first RREQ received and forwarded. The requested destination behaves similarly: it generates an RREP only

upon the first reception of an RREQ from a given source. Routes discovered in this mode may be suboptimal, but they are acquired with minimal delay.

Shortest path mode Routers may forward the first RREQ message received from a given source to a given destination; later arriving RREQs with the same pair (src, sequence_number) will be forwarded only if they advertise better paths than all previously forwarded RREQs with that same (src, sequence_number). The destination behaves similarly, generating an RREP for the first RREQ and then for any subsequent RREQ that advertises a better path than all previously received RREQs with that same (src, sequence_number). Depending on what metric is used for calculating "better", this can improve the quality of the acquired routes, at the cost of increasing considerably the overhead associated to route discovery processes. This is the most common mode, and used in protocols, like LOADng [20].

2.1.2. 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. On detecting that a route has failed, a route error (RERR) message is generated. On receiving such an RERR message, the source of the failed data packet can initiate a new route discovery procedure to re-establish connectivity.

2.2. Jitter Technique for Route Request (RREQ) Flooding

Simultaneous packet transmissions and, in particular, those performed in reactive protocols during route discovery processes, are likely to cause packet losses in wireless mesh networks, due to collisions between concurrent transmissions of routers having (at least) a common neighbor. In order to prevent or minimize these collisions, RFC 5148 [9] recommends the use of jitter for different cases in which packets may be expected to be sent concurrently. Several well-known reactive protocols (e.g., AODV [18], LOAD [22], LOADng [20]) use or provide support to jitter when flooding RREQ packets over a wireless sensor network.

Without jitter, a router receiving an RREQ packet to be forwarded retransmits it immediately after processing. As shown by the example in Figure 1, because retransmissions in neighboring routers are triggered by this single event (the reception of the RREQ packet at *B* and *C*), there is a high probability of collision. Instead, when using jitter, every receiving router adds a small, random delay before rebroadcasting the RREQ packet, which dramatically reduces the cases of packet collisions (see Figure 1c).

RFC 5148 [9] recommends that delays are selected following a uniform distribution between zero and a maximum jitter value, J_m. Note that this is the maximum entropy distribution among those assigning continuous jitter values between zero and J_m [23]; the use of this distribution thus maximizes the randomness of the total delay that is incurred by an RREQ packet sent along a certain path.

Figure 1. Timeline for packet flooding from A: (**a**) Network topology; (**b**)With direct retransmission by B and C; (**c**) With retransmission by B and C after a random delay. The network topology contains four routers $\{A, B, C, D\}$ and four bidirectional links $\{AB, AC, BD, CD\}$.
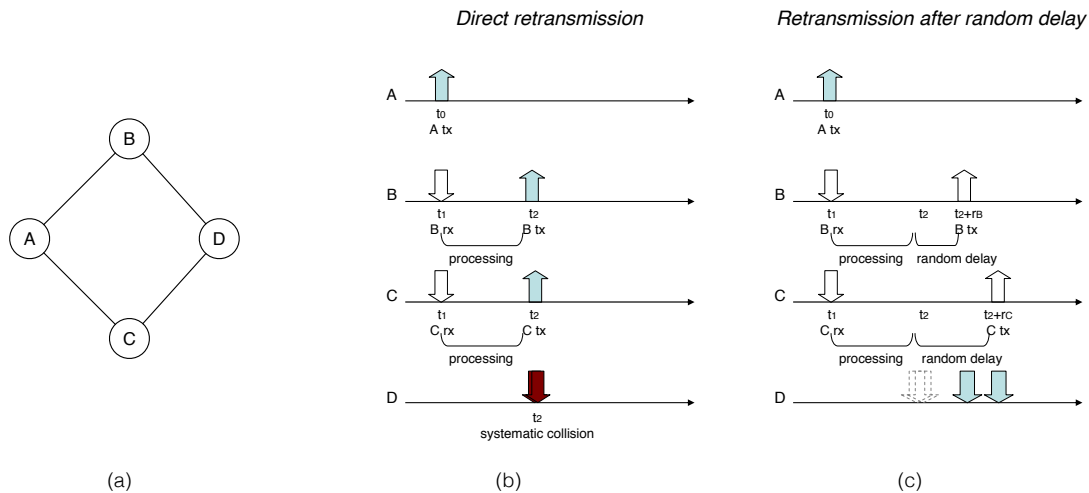
Other than the prevention of packet collisions from simultaneous transmissions, the use of jitter in flooding has two immediate additional effects:

1. The RREQ flooding, and, therefore, the route discovery, is slowed; and

2. Routers need larger buffers to store packets that have been received, but not yet forwarded.

The trade-off between these drawbacks and the reduction in the probability of collisions, when jitter is uniformly distributed (according to RFC 5148 [9]), can be controlled by way of the length of the jitter interval, J_m [15].

3. Delay Inversion and Jitter Optimizations

This section analyses the delay inversion effect, a side-effect of applying uniformly distributed jitter on route request (RREQ) flooding. In order to counter the delay inversion effect, an approach called "window jitter" is introduced, in Section 3.2, applicable for networks using a simple hop count metric. Of course, hop count metrics have their own issues, notably the worst-path-first syndrome, and therefore, Section 3.3 presents a generalization of "window jitter" to non-trivial link metrics.

3.1. The Delay Inversion Effect

Consider the topology shown in Figure 2a and assume that router A floods an RREQ, in order to discover a route towards D. A "trivial" hop count metric is used in this example, *i.e.*, the path with less hops is preferred. Absent jitter, the RREQ would reach D through the path $p_2 = \{A, E, D\}$ faster than it would through the path $p_1 = \{A, B, C, D\}$; assuming that processing time at each intermediate router, before retransmission, is similar.

Now, if a random delay on retransmission—jitter—is applied at each hop and if that jitter is selected from a uniform random distribution $[0, J_m]$, the message copy sent through the longer path (in terms of

the number of hops), p_1, may reach the destination faster than the message copy over p_2; and this, with a non-negligible probability. Figure 2b illustrates this case, and Section 4.1.1 analyzes the probability in further detail.

With reference to Figure 2, consider the transmission of an RREQ packet from A, received simultaneously at B and E. Although the RREQ needs to traverse two hops (B and C) to reach D via p_1 and only one hop (E) via p_2, the RREQ sent across p_1 may be received first at D if $j_E > j_B + j_C$, as shown in Figure 2b.

Router D replies to the route request from A with an RREP that advertises the (longer-than-shortest) path p_1. When the RREQ, traversing p_2, reaches D, D replies by generating another RREP that advertises the (shorter) path p_2. This implies that A gets, and possibly uses for a certain amount of time, a suboptimal path towards D (p_1), and it needs to receive two RREP from D in order to learn the optimal path from A to D. If D is not the ultimate destination sought by A, then D would retransmit two copies of the RREQ—one received via each of the paths p_1 and, then, p_2.

Figure 2. Delay inversion example. Router A tries to broadcast an route request (RREQ) message through the network, through paths p_1 and p_2. The RREQ through the longer path $\{A, B, C, D\}$ travels faster than the one through the shorter path $\{A, E, D\}$, due to (unfortunate) jittering. (**a**) Topology; (**b**) Example of jitter value assignment for an RREQ from A towards D.

(**a**)   (**b**)
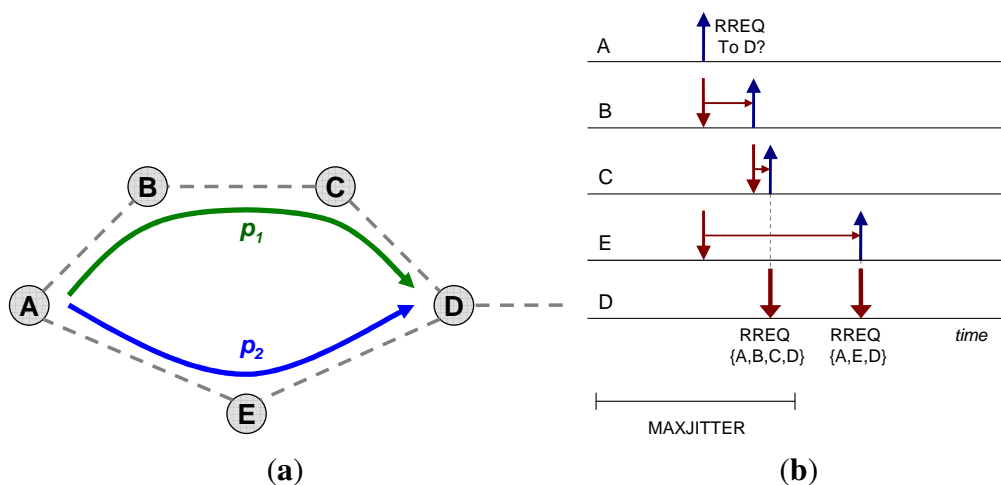
This example illustrates that the use of a uniform random distribution for jitter values when forwarding RREQ packets during route discovery in a reactive routing protocol may lead to cases in which "transmissions over longer paths get first". This effect is hereafter denominated delay inversion caused by jitter.

Delay inversion is harmful due to at least three undesirable effects:

(i) It increases the (probability of) sub-optimality of reported routes;

(ii) It increases the impact of data traffic forwarded through the network, as a consequence of the use of suboptimal routes; and

(iii) It increases the amount of control traffic: duplicate RREQs forwarded and multiple RREPs generated.

3.2. Window Jitter

Window jitter is a small modification of the uniform distribution of jitter, as recommended by RFC 5148 [9]. It introduces a minimum amount of jitter that must be incurred in each hop. Jitter values are, then, instances of a random variable $TJ_W \sim$ Uniform $[\alpha J_m, J_m]$, where $\alpha \in (0, 1)$ and αJ_m is a minimum jitter value. Note that $\alpha = 0$ corresponds to the uniform jitter distribution specified in RFC 5148 [9]; $\alpha = 1$ would imply a deterministic delay (of length J_m). The fact that $\alpha \neq 0$ entails that the lower bound for the RREQ delay grows linearly with the length of the traversed path.

Window jitter reduces the randomness and increases the (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 J_w, the total delay caused by jitter in a path of n hops belongs to the interval $[n\alpha J_m, nJ_m]$ ($\alpha \neq 0$). The trade-off between randomness and path length deterministic dependence can be controlled by way of parameter $\alpha \in (0, 1)$: the closer α is to one, the more deterministic becomes the total delay of an RREQ packet with respect to the path length.

Under the window jitter distribution, each additional hop in the path traversed by an RREQ packet causes at least an additional delay of αJ_m. Intuitively (see Section 4.1.2 for a more rigorous analysis), this makes it less likely that a larger number of hops is traversed by RREQ packets in a shorter time. The implicit assumption is that "shorter" paths (in the number of hops) are preferable to "longer" paths, considered worse for routing (hop count metric).

3.3. Adaptive Jitter for Non-Trivial Metrics

The window jitter principle can be naturally extended to non-trivial link metrics, for instance based on the probability of successful transmission (Expected Transmission Count [24]) or the available bandwidth in the link. This extension of window jitter to link metrics other than hop count is denominated adaptive jitter.

Given a link quality indicator LQ $\in (0, 1)$ (LQ $\longrightarrow 1$ for high quality links), jitter values are selected uniformly within the interval $[(1 - LQ)J_m, J_m]$. This reduces the probability of delay inversion or, equivalently, increases the probability that an RREQ packet is forwarded faster by routers receiving it on better links.

Note that the window jitter distribution presented in Section 3.2 corresponds to the particular case of LQ $= 1 - \alpha$ for all available links.

4. Analysis

This section examines, analytically, the impact of jitter being applied to RREQ flooding in reactive routing protocols. Two analytic models of jitter distributions are considered: uniform jitter, as specified in RFC 5148 [9], and window jitter as proposed in Section 3.2, for a static α. Specifically, Section 4.1 studies the quality of routes discovered throughout the network and the probability that the delay inversion effect occurs when two paths of a different number of hops are available between the requesting source and the requested destination. Following, Section 4.2 proposes simplified models for examining and comparing other side-effects of each uniform and window jitter, in particular, the probability of

auto-collisions—two "copies" of the same packet, occurring in the network due to retransmissions and colliding—when flooding an RREQ, and the average number of received-but-not-yet-forwarded RREQs in a router. Appendix A contains the full proofs of the propositions in this section.

4.1. Route Sub-Optimality

This section provides a quantitative probabilistic analysis of the delay inversion effect. Let T_j be the random variable for jitter values. The delay caused by jitter in an RREQ message traversing a path of n hops, $T^{(n)}$, can be then be computed as follows:

$$T^{(n)} = \sum_{i=1}^{n} T_{j_i} \tag{1}$$

Given two paths between a source X and a destination Y, with lengths n and m, let $D^{(n,m)}$ 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, respectively. It is a random variable that depends on the random variables for the jitter values in the way shown in Equation (2):

$$
\begin{aligned}
D^{(n,m)} &= T^{(n)} - T^{(m)} = \sum_{i=1}^{n} T_{j_i} - \sum_{j=1}^{m} T_{j_j} = \\
&= \sum_{i=1}^{n} T_{j_i} + \sum_{j=1}^{m} \left(-T_{j_j} \right)
\end{aligned} \tag{2}
$$

The probability that the delay inversion effect occurs in the RREQ flooding corresponds to $Pr(D^{(n,m)} > 0|_{n<m})$, whose expression is detailed in Equation (3):

$$P\left(T^{(m)} < T^{(n)}\right) = P\left(D^{(n,m)} > 0\right) = \int_0^\infty f_{D^{(n,m)}}(t)dt \tag{3}$$

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

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

where $\otimes$ denotes the convolution.

4.1.1. Uniform Jitter

In the case of uniform jitter, $T_j \equiv \text{Uniform}\,[0, J_m]$. Let P_U and D_U denote the probability of inversion with uniform jitter and the inter-path delay difference with uniform jitter, respectively. Then, the probability of inversion P_U has the expression detailed in Proposition 1.

Proposition 1. *The probability of inversion with uniform jitter, P_U, in two paths of length n and m, has the following expression:*

$$P_U = \frac{1}{(n+m)!} \sum_{k=0}^{n+m} \left[(-1)^k \binom{n+m}{k} \times \left((n+2m-k)^{n+m} - (m-k)_+^{n+m} \right) \right] \tag{5}$$

Figure 3a 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 3b 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. Figure 4 illustrates the evolution of the probability of delay inversion between paths with a constant hop difference $\Delta = m - n$ as the absolute length of these paths (in number of hops) increases. Note that both Figures 3b and 4 show bidimensional cuts of the surface presented in Figure 3a; these cuts result from the intersection of this surface with planes $\pi_1 : \{m = ct.\}$ and $\pi_2 : \{\Delta = m - n = ct.\}$, respectively.

Figure 3. 3D and 2D representation of P_U. (**a**) Theoretical value of $P_U \equiv Pr(D_U^{(n,m)} > 0)$, for $1 \leq n \leq 6$, $1 \leq m \leq 6$; (**b**) Restriction of P_U for $m = 1, 2, 3, 4, 5$, $n < m$ (theoretical values in lines, simulations in points).

(**a**)   (**b**)
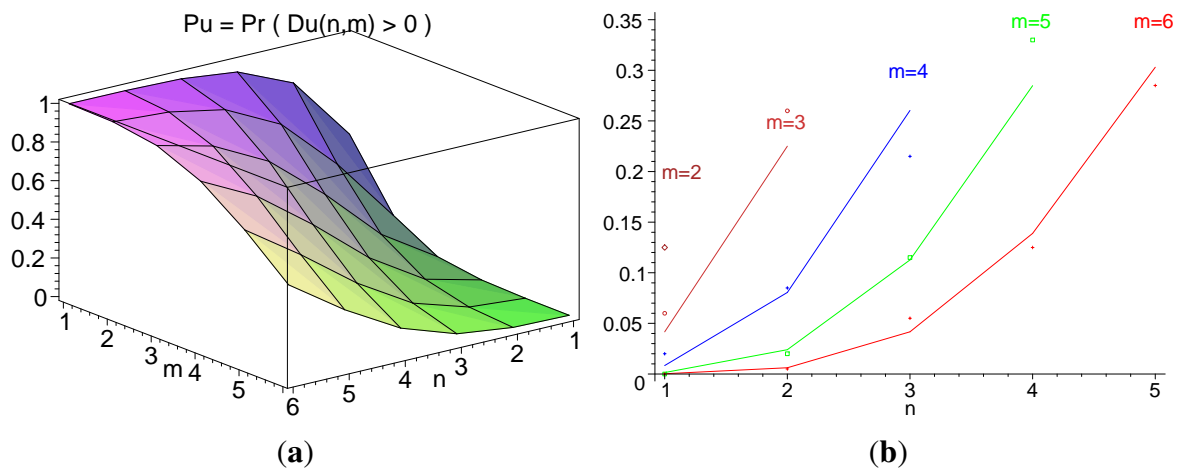
Figure 4. Value of $P_U \equiv Pr(D_U^{(n,m)} > 0)$, depending on $\Delta = m - n > 0$, for different values of n.
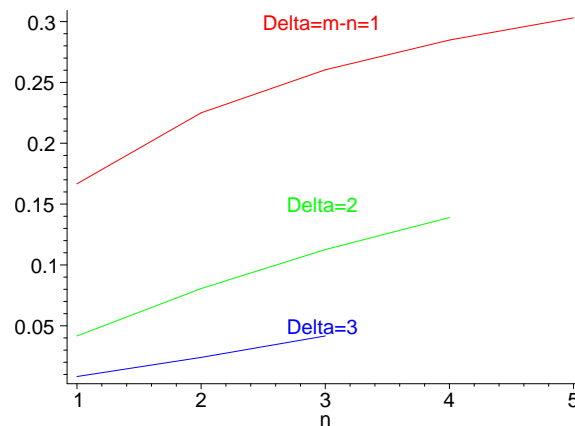
Expression (5) indicates that the delay inversion occurs, under the conditions specified in RFC 5148 [9], with a significant probability. For the topology presented in Figure 2a, for instance ($n = 2$, $m = 3$), this probability is $Pr(D_U^{(2,3)} > 0) = 0.225$, implying that the RREQ traversing the longer path will reach the destination first in almost one out of four cases.

Two aspects can be highlighted from this analysis: (i) from Equation (5), 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 3b and 4, the 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.

The fact that delay inversion is more frequent in long paths (in terms of number of hops) is due to the fact that the range in which total jitter values are possible (adding all per-hop jitter values) has a linearly growing upper bound (nJ_m, where n is the path hop length) and a fixed lower bound set to zero. Recall that, while a single uniform jitter value within $[0, J_m]$ has mean $\frac{J_m}{2}$ and variance $\frac{J_m^2}{12}$, the sum of n uniform jitter values converges in law (by using the Central Limit Theorem, for $n \longrightarrow \infty$) to a Gaussian distribution with mean $n\frac{J_m}{2}$ and variance $n\frac{J_m^2}{12}$: longer paths (with n hops) thus lead to more variant total jitter distributions.

4.1.2. Window Jitter

The analysis performed in Section 4.1.1 for $T_j \equiv$ Uniform $[0, J_m]$ can be repeated for $T_j^* \equiv$ Uniform $[\alpha J_m, J_m]$. The probability density distribution (pdf) of the inter-path delay difference described in Equation (4) thus becomes, for the case of window jitter, $D_W^{(n,m)}$:

$$f_{D_W^{(n,m)}}(t) = \left(\bigotimes_{i=1}^{n+m} f_{T_j^*{}_i} \right) (t + mJ_m) \tag{6}$$

Without loss of generality, and for the sake of simplicity, it can be assumed in the following that $J_m = 1$. Then, also using Equation (A1), the pdf detailed in Equation (6) can be expressed in the terms of Proposition 2.

Proposition 2. *The probability density function of the inter-path delay difference for window jitter, $f_{D_W^{(n,m)}}(t)$, has the expression shown in Equation (7):*

$$f_{D_W^{(n,m)}}(t) = \frac{1}{1-\alpha} \frac{1}{(n+m-1)!} \sum_{k=0}^{n+m} (-1)^k \left[\binom{n+m}{k} \times \left(\frac{t - n\alpha + m}{1-\alpha} - k \right)_+^{n+m-1} \right] \tag{7}$$

where $(.)_+$ notation stands for the positive part (i.e., $z_+ = z$ if $z \geq 0$, and zero otherwise), for $\frac{t - \alpha n + m}{1-\alpha} \in [0, n+m]$, and zero otherwise.

Therefore, the probability of delay inversion with paths of length n and m, $Pr(D_W^{(n,m)} > 0)|_{n<m}$, can be computed as indicated in Proposition 3.

**Proposition 3.** *The expression of $P_W \equiv Pr(D_W^{(n,m)} > 0)$ corresponds to:*

$$\begin{aligned} P_W = \ & \frac{1}{(n+m)!} \left(\sum_{k=0}^{n+m} (-1)^k \binom{n+m}{k} (n+m-k)^{n+m} - \right. \\ & \left. - \sum_{k=0}^{k^*} (-1)^k \binom{n+m}{k} \left(\frac{m - n\alpha}{1-\alpha} - k \right)^{n+m} \right) \end{aligned} \tag{8}$$

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

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

It can be observed in Figure 5a that the transition from $P_W = 0$ to $P_W = 1$ (*i.e.*, from situations in which RREQ transmissions over the n-path are never faster than those over the m-path, to a situation in which they are always faster) is significantly more steep with the modified (generalized) distribution of jitter values than with the distribution of RFC 5148 [9] (see Figure 3a). 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.

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

(**a**) (**b**)
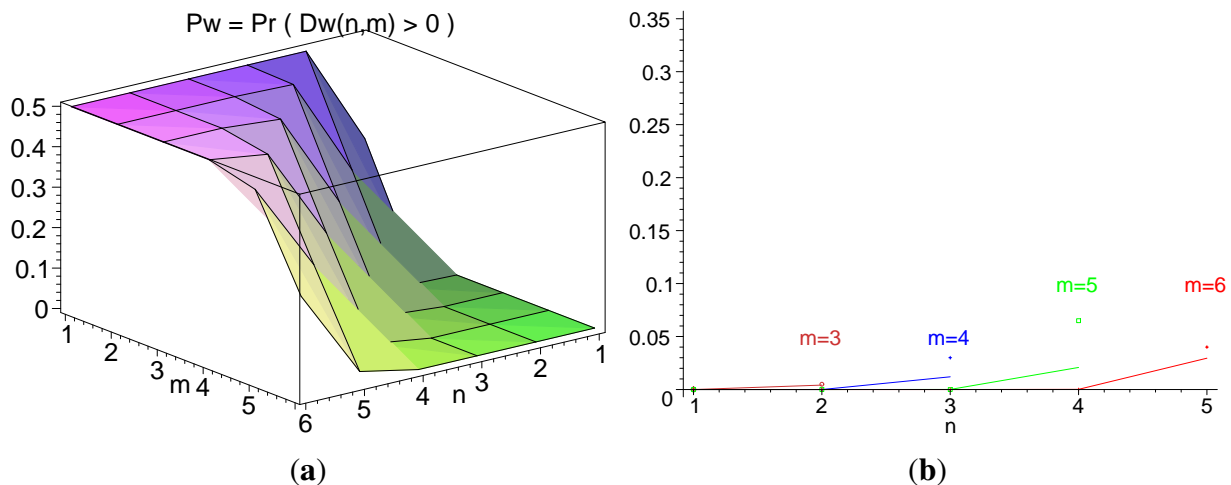
The adaptive jitter for non-trivial metrics is a generalization of window jitter. Note that, under the assumption of the hop count metric (*i.e.*, link quality is constant), in this analysis, window jitter is equivalent to adaptive jitter. Further simulations, presented in Section 5, show experimentally the advantage of adaptive jitter with non-trivial metrics.

Figure 5b 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 3b, 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 $T_j \sim \text{Uniform}[0, J_m]$: for very similar ($\Delta = m - n = 1$, which is the most frequent case) and long paths ($n = 5$), the probability reduces in a factor of five and stays below 6%; the relative variation becomes still more significant as paths are shorter. The same conclusion can be drawn from the evolution of P_W with respect to $\Delta = m - n$ (Figure 6).

Figure 6. Value of $P_W \equiv Pr(D_W^{(n,m)} > 0)$, depending on $\Delta = m - n > 0$, for different values of n ($\alpha = \frac{1}{2}$). Traces corresponding to $\Delta = 2$ and $\Delta = 3$ are significantly lower than 0.01 and, thus, are not visible in the picture.
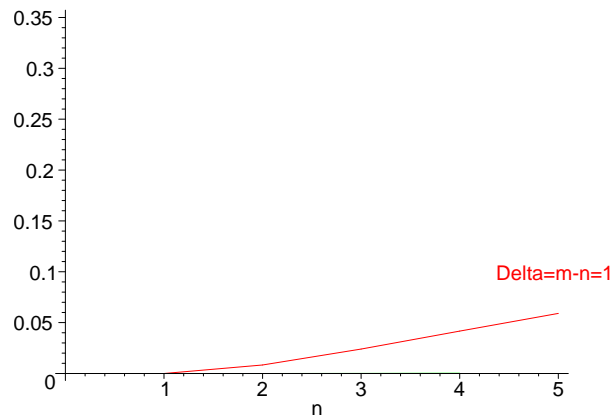
4.2. Other Effects

This section examines some additional effects of packet jittering. The impact of standard (uniform) and window jitter in terms of flooding performance and networking requirements is addressed by way of simplified models for flooded packet auto-collisions (Section 4.2.1) and buffering needs in forwarding routers (Section 4.2.2).

4.2.1. Auto-Collisions in First Hop

For a packet p flooded over a wireless network, auto-collision is the case of two copies of that packet, p, colliding; in other words, that p "collides with its own alter-ego".

Consider the network, depicted in Figure 7, in which a router S floods a packet p at time $t = t_0$ and in which all of N_1, N_2, ..., N_n are neighbors. This packet is, then, simultaneously received by the n neighbors of S, $N(S) = \{N_1, N_2, ..., N_n\}$ and forwarded (retransmitted) with a random jitter value T_j, distributed according to the different considered mechanisms. For tractability purposes, retransmissions are modeled in this section as a homogeneous Poisson arrival process. In this scenario, there is an auto-collision of p in the one-hop neighborhood of S when at least two neighbors of S retransmit p at times t_1 and t_2, such that the inter-arrival time $|t_1 - t_2| < L$, where $L = L(p)$ is the transmission time for packet p.

In the case of uniform jitter, retransmission times will be distributed within $t = t_0$ and $t = t_0 + J_m$, where J_m is the upper bound for jitter values; the Poisson rate of retransmissions will be $\lambda_{\text{uniform}} = \frac{n}{J_m}$. When using window jitter, retransmissions of a packet p received at $t = t_0$ will be confined between $t_0 + \alpha J_m$ and $t_0 + J_m$ ($\alpha \in (0, 1)$), that is, no retransmission will occur before $t_0 + \alpha J_m$. This implies that the Poisson rate of retransmission will grow to $\lambda_{\text{window}} = \frac{n}{(1-\alpha)J_m}$. Lemma 1 describes the probability that the flooding process of packet p leads to an auto-collision in the one-hop neighborhood of S, with uniform and window jitter.

Figure 7. Simple network topology, in which S is a neighbor with all of N_1, ..., N_n and where all of N_1, ..., N_n are neighbors of each other.
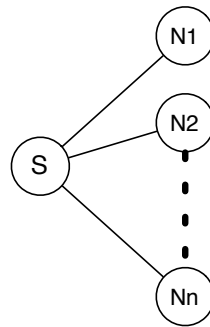

Lemma 1. *If S's neighbors use uniform jitter within $[0, J_m]$, the probability of auto-collision in the one-hop neighborhood of S corresponds to Equation (9). If window jitter is used in the same interval, with parameter α, the probability of auto-collision in the one-hop neighborhood of S corresponds to Equation (10):*

$$Pr(\text{auto-collision with uniform jitter}) = \left(1 - e^{-\frac{nL}{J_m}}\right)^n \tag{9}$$

$$Pr(\text{auto-collision with window jitter}) = \left(1 - e^{-\frac{nL}{(1-\alpha)J_m}}\right)^n \tag{10}$$

Figure 8 illustrates the probability of auto-collisions in the first hop of the flooding process (*i.e.*, when an RREQ is retransmitted by routers one hop away from the source) for different sets of parameters (n, J_m, L). Although the specific values for the probability of auto-collision depend on the chosen values, and, in particular, the relation between L and J_m, there are some general trends that can be identified. The use of window jitter leads to more frequent auto-collisions at the first hop, due to the effective reduction of the interval of retransmissions (from $[0, J_m]$ to $[\alpha J_m, J_m]$), which implies a more deterministic retransmissions schedule by one-hop neighbors of S. This effect, however, is specific to the first hop and becomes less relevant in multi-hop flooding, as the network has a bigger diameter and routes grow longer, as shown in further experiments (Sections 5 and 6).

Figure 8. Probability of auto-collision in the first hop.
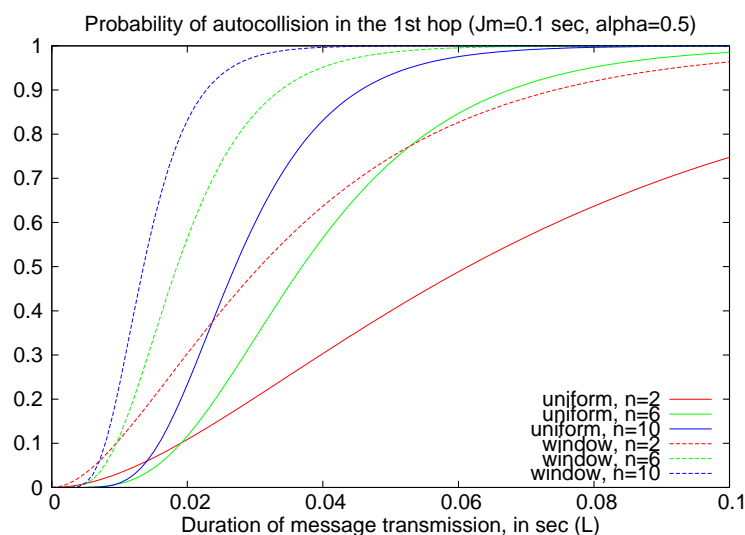
4.2.2. Buffer Occupancy

When using jitter in a flooding operation, the flooded packets must be stored in buffers in the intermediate routers for the time imposed by the jitter interval. The average occupancy of intermediate routers' buffers is obviously related to the distribution of jitter values.

This section computes the average number of packets to be forwarded by a particular router at time $t = t_0$, for each of the different jittering distributions studied. Assuming a Poisson process with rate λ for packet arrival, the number of packets that arrive (on average) within $[t_0 - J_m, t_0)$ is λJ_m. The probability that one of these is scheduled to be sent later than t_0 is $Pr(\bar{T}_t + T_j > t_0)$, where $\bar{T}_t$ stands for the random variable of the arrival time (uniformly distributed within $[t_0 - J_m, t_0)$, according to the Poisson model) and T_j is the random variable for jitter value.

Then, the average buffer occupancy B has the following expression:

$$B \;=\; \lambda J_m Pr(\bar{T}_t + T_j > t_0) \tag{11}$$

The statistical distribution of $\bar{X} = \bar{T}_t + T_j$ is determined by the jitter value distribution:

- For the uniform jitter, $Pr(\bar{X} > t_0) = \frac{1}{2}$.

- For the window jitter ($\alpha \in (0,1)$), $Pr(\bar{X} > t_0) = \frac{1+\alpha}{2}$.

Since the use of window jitter entails, for the same value of J_m and the same flooding rate λ, a larger average delay before retransmission than uniform jitter, buffers are expected to store, on average, a α additional fraction of packets to be forwarded.

5. Graph Simulations and Results

In order to evaluate the performance of the different jitter mechanisms, simulations are performed by way of a Maple-based discrete-event network graph simulator. For simplicity, simulations in this section assume $J_m = 1$ s (for all types of jitter) and $\alpha = 0.5$ (for window jitter).

5.1. Setup

The performance of the three different types of jitter ("standard" jitter, window jitter and adaptive jitter) is evaluated in the shortest-delay mode and shortest-path mode of RREQ flooding (see Section 2.1.1) for different network graph scenarios. The different network scenarios are defined by triplets (N, ρ, metric), where:

- N is the network population (number of routers in the graph);

- ρ is the network router density (number of routers per km^2); and

- "metric" identifies the link metric model: uniform (hop count, in which all available links have cost one) or random (links have a random integer cost from one to 10).

Values for each network profile are averaged over 20 samples, each sample corresponding to a random (static) distribution of routers over the network grid, in which RREQs are sent from a fixed random source to a fixed random destination. Each value related to a distribution corresponds to the average (μ) of 10 RREQ flooding procedures simulated between the source and destination; standard deviation (σ) intervals around the mean value ($\mu - \sigma, \mu + \sigma$) are also displayed.

The following metrics are used for evaluating the performance of different jitter mechanisms:

- **Number of (auto-)collisions** Consistently with the definition provided in Section 4.2.1, an auto-collision (denoted as a collision in this section) is counted when a router receives two retransmissions (copies) of the same RREQ, simultaneously.

- **Optimality index** Given a source s and a destination d, the optimality index for a path between s and d is the cost of the shortest path divided by the cost of the discovered path. The path with the index closer to one indicates a better path.

- **Routing overhead** Measured in the number of RREQ or RREP retransmissions.

- **Route discovery delay** In shortest-delay mode, this is the time required in order to discover the first path. In shortest-path mode, it is the time required in order to discover the best path.

5.2. Results

This section presents the results of the simulation-based evaluation of "standard" jitter (*i.e.*, uniform jitter according to RFC 5148 [9]), window jitter and adaptive jitter; this for each of the different route discovery modes (shortest-delay and shortest-path), and for different families of network scenarios (fixed grid and constant density) and link metrics (uniform and random link metrics).

5.2.1. Uniform Link Metrics

The simulation of the shortest-path mode of route discovery in networks with uniform link cost (hop count) shows that window jitter is able to significantly reduce the number of collisions caused by RREQ flooding, when compared to "standard" jitter. Figure 9 depicts that the collision reduction becomes more relevant as the network density grows.

This reduction is due to the fact that the use of window jitter, when compared to "standard" jitter, increases the probability that the first RREQ received by an intermediate router (or a destination) has traversed the shortest path (according to the metric in use) available, and therefore, no additional RREQ retransmissions need be performed (and no additional route replies need be sent after the first) over a path with better quality than the one previously advertised. The better the quality of the first advertised path, the less RREP control packets involved in a single route discovery process; and the less likely the packet collisions.
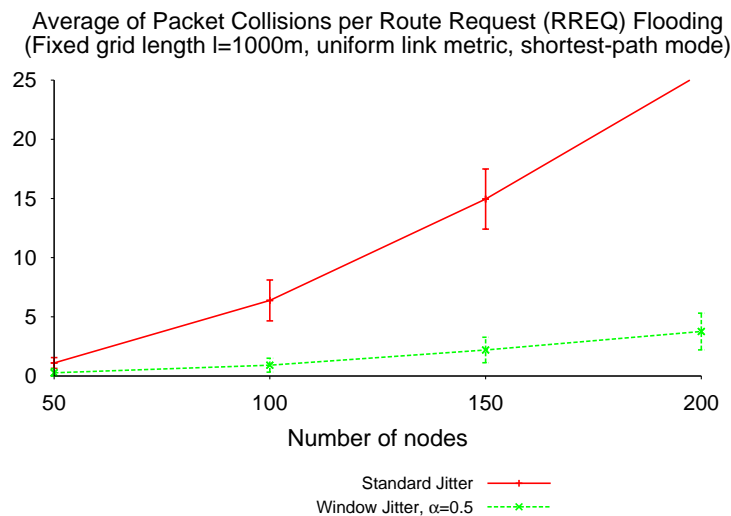
Figure 9. Number of collisions, shortest-path RREQ (Route Request) flooding.

Figure 10 depicts the optimality index of window jitter and "standard" jitter, as a function of network density, when using shortest-delay RREQ flooding. When routers are only allowed to forward the first RREQ received from a given source towards a given destination, the use of window jitter improves significantly the quality of the routes identified through RREQ flooding. This confirms the results from the theoretical analysis of Section 4.1 about the probability of delay inversion in "standard" jitter and window jitter.

Figure 10. Optimality index, shortest-delay RREQ flooding.
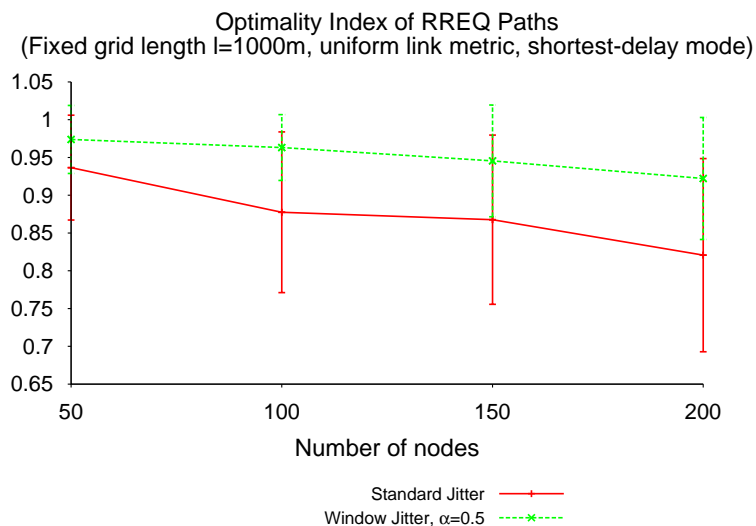
As mentioned in Section 3.2, the objective for window jitter is to ensure that the first RREQ that is received, also is the RREQ that has traversed the fewest number of hops and, therefore, represents the better path. Window jitter implicitly assumes a constant link metric, and it is able to provide a significant improvement in the route discovery performance when no more information about link quality is available.
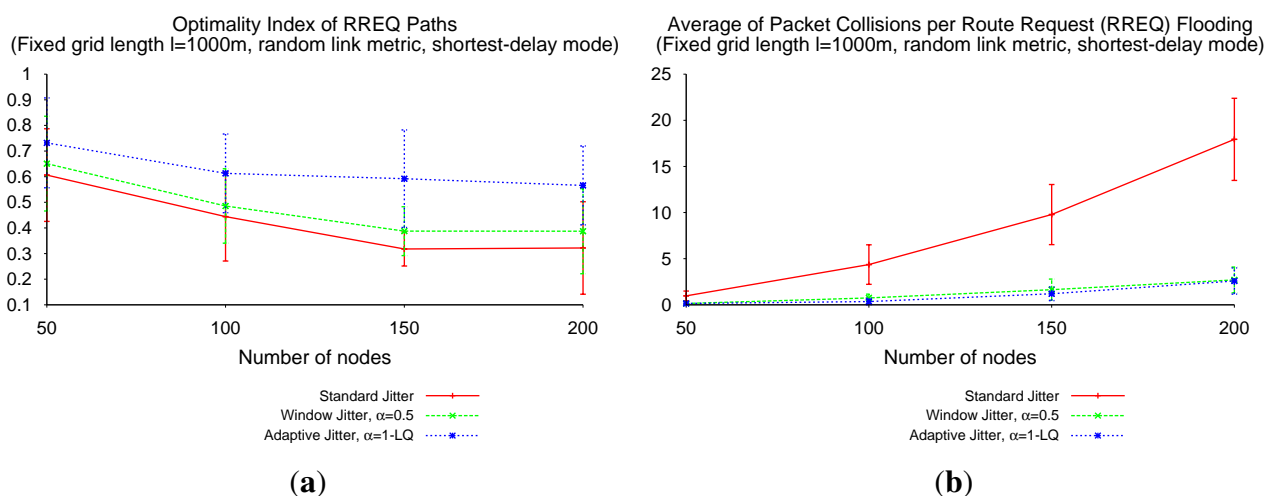
5.2.2. Shortest-Delay Mode over Non-Trivial Link Metrics

The advantages of window jitter over "standard" jitter is less significant when link metrics are not uniform: the ability to identify better paths by introducing fixed minimum delays (αJ_m) per hop degrades, as depicted in Figure 11. For these non-trivial link metrics, the simulated results show that the use of the adaptive jitter presented in Section 3.3 is more adequate. This is because routers using adaptive jitter can take the actual link metric (e.g., ETX, bandwidth, *etc.*) into consideration, rather than the single presence of these links in the path.

Figure 11a depicts the fact that adaptive jitter clearly outperforms window jitter and "standard" jitter in terms of optimality index. As depicted in Figure 11b for random link quality values, this benefit from the adaptive jitter is compatible with a low level of packet collisions (similar to the level achieved with window jitter and significantly lower than the level achieved with "standard" jitter) in networks with heterogeneous link qualities (*i.e.*, non-uniform metrics).

Discrimination of RREQs based on the quality of traversed links is performed by introducing pre-forwarding delays. This entails a trade-off between RREQ path optimality and RREQ flooding delays, as depicted in Figure 12, for the three types of jitter. In general, the better the path indicated in the first RREQ received by the intended destination, the more delay between the RREQ transmission by the source and its reception in the destination. This can be observed, in particular, for networks of constant router density (Figure 12b). Results from Figure 12a indicate, in addition, that additional delay caused by adaptive jitter with respect to window jitter strongly depends on the network density: as more paths are available to reach the destination (because the network is denser), the heterogeneity of the quality of the involved links in flooding is also higher and the adaptive jitter type allows one to deliver route requests (RREQs) faster, while window jitter cannot reduce the per-hop delay beyond a minimum value αJ_m.

Figure 11. Shortest-delay RREQ flooding. (**a**) Optimality index; (**b**) RREP (route reply) packet collisions.

(**a**) (**b**)

Figure 12. Route discovery delay, shortest-delay RREQ flooding. (**a**) Fixed grid; (**b**) Constant router density.


(**a**)

(**b**)
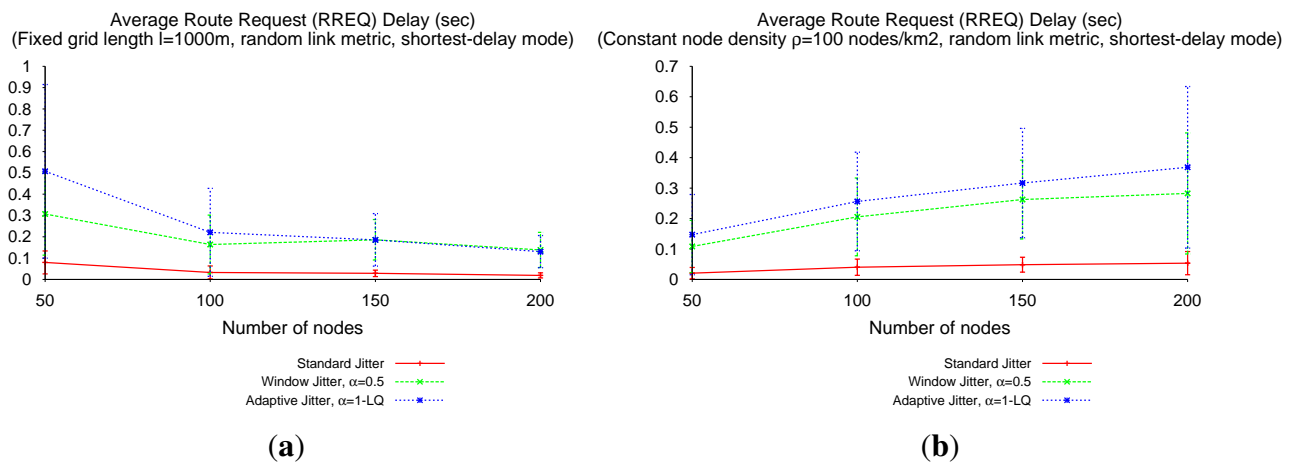
5.2.3. Shortest-Path Mode over Non-Trivial Link Metrics

The use of adaptive jitter in the shortest-path mode of route discovery is also beneficial, although not due to the same reasons (RREQ path quality improvement, mainly) as in the shortest-delay mode. The fact that routers are able to forward RREQs indefinitely, any time that they receive an RREQ with a better route than the last forwarded RREQ, entails that RREQ flooding ideally provides the optimal route between source and destination, if it terminates successfully (without packet losses, collisions or inaccuracies in link quality estimation). However, the shortest-path mode with static jitter ("standard" jitter, window jitter) presents a relevant drawback: as every packet may forward each RREQ several times, and the source may send several RREP to the same destination, the probability of packet collisions and route discovery failure also increases; more significantly for dense networks. Figure 13a,b depicts the evolution of RREQ retransmissions and RREP transmissions per route discovery, when the network density increases. It can be observed that the use of adaptive jitter, by increasing the quality of the firstly-discovered paths, entails a reduction in the number of control packets per route discovery (RREQ retransmissions and route replies) up to 30%, with respect to the static configurations.

Figure 14 depicts average RREQ delays for the different types of jitter when using shortest-path (sh-p) and shortest-delay (sh-d) modes. For any given type of jitter, the delay for the shortest-path mode is always longer or equal to the delay for the shortest-delay mode: in the later, the flooding terminates when the destination receives the first RREQ; in the former, the flooding terminates when the destination receives the RREQ through the best path, which can correspond to the first or to a posterior reception. More interestingly, two observations can be drawn from Figure 14. In the first term, RREQ delay caused by adaptive jitter decreases with the network density (a result consistent with what was depicted in Figure 12), while, in contrast, "standard" jitter and window jitter are present in the shortest-path mode, a roughly constant delay with respect to network density. In the second term, the gap between RREQ delays in shortest-path and shortest-delay modes, *i.e.*, the additional delay caused by reception in the destination of better RREQ packets later than the first, is different for each type of jitter. The adaptive type of jitter has the smallest gap between modes, which is consistent with the previous observation about the quality of first-received RREQs at the destination. The significant difference between modes

when using window jitter is another indication, in turn, of the poor performance achieved by this type of jitter in networks with diverse link qualities, as the non-trivial link metrics scenarios considered in this section.

Figure 13. Route overhead per route discovery, shortest-path RREQ flooding. (**a**) The number of RREQ (re)transmissions; (**b**) The number of RREP (re)transmissions.

(**a**)

(**b**)
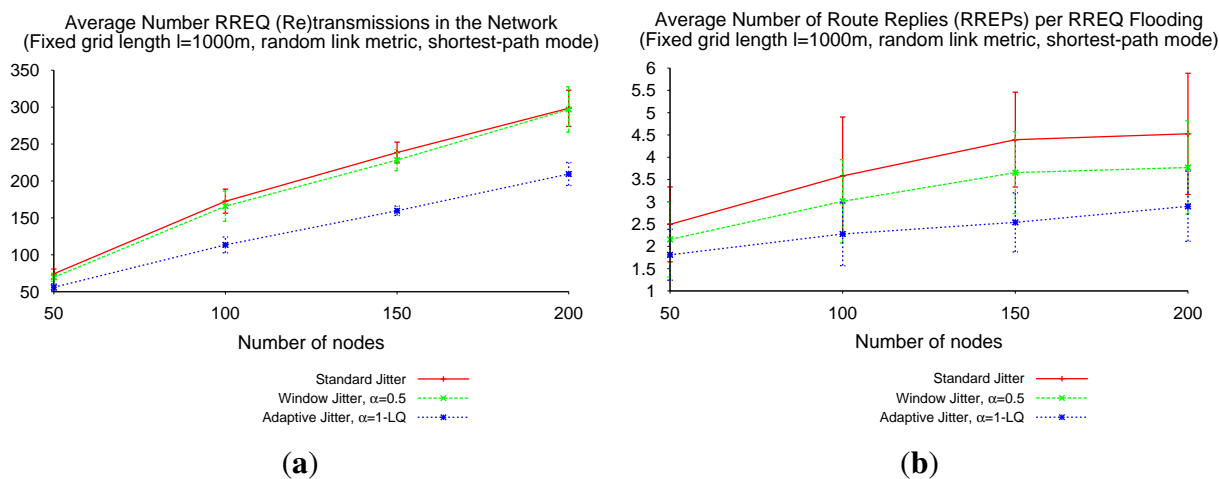
Figure 14. Route discovery delay, shortest-path and shortest-delay RREQ flooding.
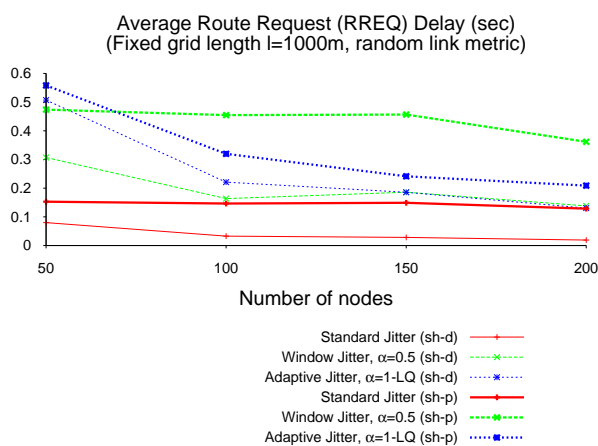
6. Network Simulation and Results

Window jitter, proposed in Section 3.2, is implemented, studied and evaluated by way of network simulations of several different network scenarios and parameter configurations. These simulations permit validating the theoretical results obtained in Section 4, as well as the algorithmic evaluation described in Section 5, by considering all network layers, including the transport layer, MAC (Medium Access Control) layer and the physical propagation model.

This section compares the performance of LOADng RREQ flooding when using the original "standard" jitter (*i.e.*, uniform jitter according to RFC 5148 [9]) and the proposed window jitter. Simulations also allow one to identify the networking and jitter elements that have an impact on this performance. Section 6.1 presents the setting of the performed network simulations. Section 6.2 describes the main results obtained in the experiments.

6.1. Setting

NS2 (Network Simulator 2) simulation results are presented in Section 6.2. Simulations were made of a field of 1000×1000 m, with varying numbers of routers placed randomly, equipped with a 802.11b radio. For the purpose of this study, router mobility was not considered. The metric is based on hop-count.

Each simulation lasts for 100 s. 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, $J_m = 100$ ms. Jitter is selected within $[0, 100]$ ms (mean, 50 ms).

- Standard RFC 5148 jitter, $J_m = 200$ ms. Jitter is selected within $[0, 200]$ ms (mean, 100 ms).

- Window jitter, $\alpha = \frac{1}{2}$, $J_m = 100$ ms. Jitter is selected within $[50, 100]$ ms (mean, 75 ms).

- Window jitter, $\alpha = \frac{2}{3}$, $J_m = 150$ ms. Jitter is selected within $[100, 150]$ ms (mean, 125 ms).

Figure 15 shows the probability density function (pdf) for the different jitter settings.

Figure 15. Pdfs for the four examined jitter settings.
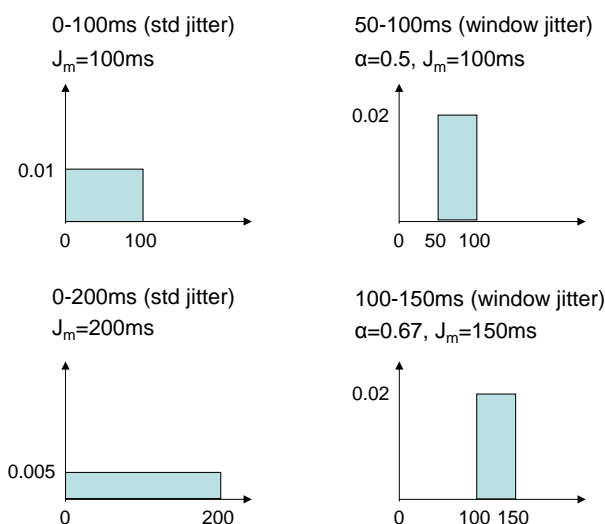
For each of the five settings, the shortest-path and shortest-delay strategies for the RREQ forwarding scheme are evaluated.

6.2. Results

This section describes the most relevant results, observed in the performed simulations of various jitter distributions (no jitter, "standard" jitter and window jitter) under shortest-path RREQ forwarding (Section 6.2.1) and shortest-delay RREQ forwarding (Section 6.2.2) configurations.
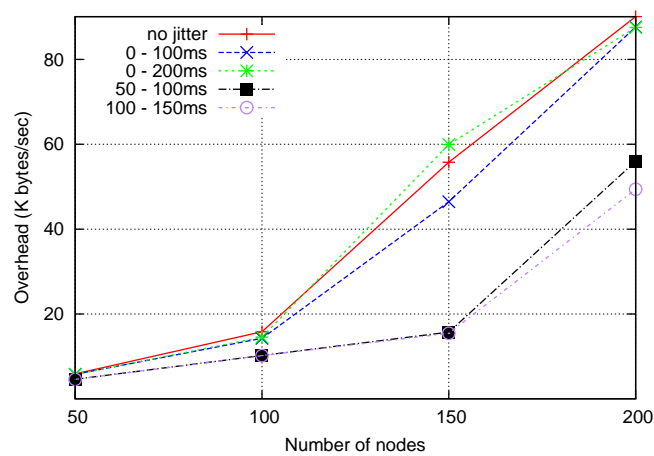
6.2.1. Shortest-Path RREQ Forwarding

The first observation that can be highlighted from the shortest-path RREQ forwarding results is that the use of "standard" jitter does not have an significant 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 and "standard" and no jitter.

Figure 16a shows the number of collisions with a different density of routers. When using no jitter, the highest number of collisions, slightly higher than with "standard" jitter, occurs. This is because adjacent routers are more likely to retransmit received RREQs at the same time. Window jitter (50–100 ms, 100–150 ms) yields significantly fewer collisions, especially in high-density scenarios. This is because the use of the window jitter enables forwarding routers to reduce the number of transmissions (*i.e.*, overhead) by reducing the cases of delay inversion, as shown in Figure 16b.

Figure 16. Simulation results of shortest-path RREQ forwarding. (**a**) Number of collisions; (**b**) Average overhead.
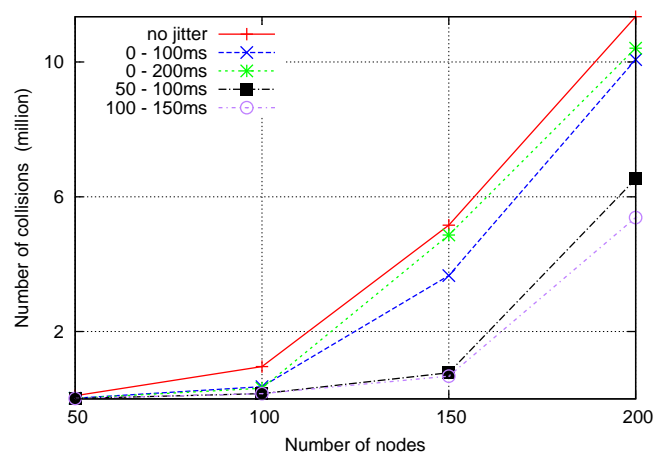

(**a**)

(**b**)

Regarding 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, depicted in Figure 17b, it is interesting to observe that the impact of the different types of jitter is different for low-density and high-density networks. The average route discovery delay for each type jitter is highly correlated with the mean of the random jitter variable: settings with higher jitter means exhibit higher average delays in the route discovery processes. For denser networks, in contrast, window jitter presents better performance in terms of route delivery delay than "standard" jitter, regardless of the mean value of the jitter random variable. The use of "standard" jitter (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, as depicted in Figure 16a , and the data packet delivery ratio, as depicted in Figure 17a. 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 17c) and allows one to reduce the number of RREPs sent in response to an RREQ. This alleviates the control traffic load of the network and decreases the number of packet collisions, therefore significantly reducing the average delay for route discovery processes.

Figure 17. Simulation results of shortest-path RREQ forwarding. (**a**) Data packet delivery ratio; (**b**) The average route discovery delay; (**c**) Average path length.


(**a**)

(**b**)

(**c**)
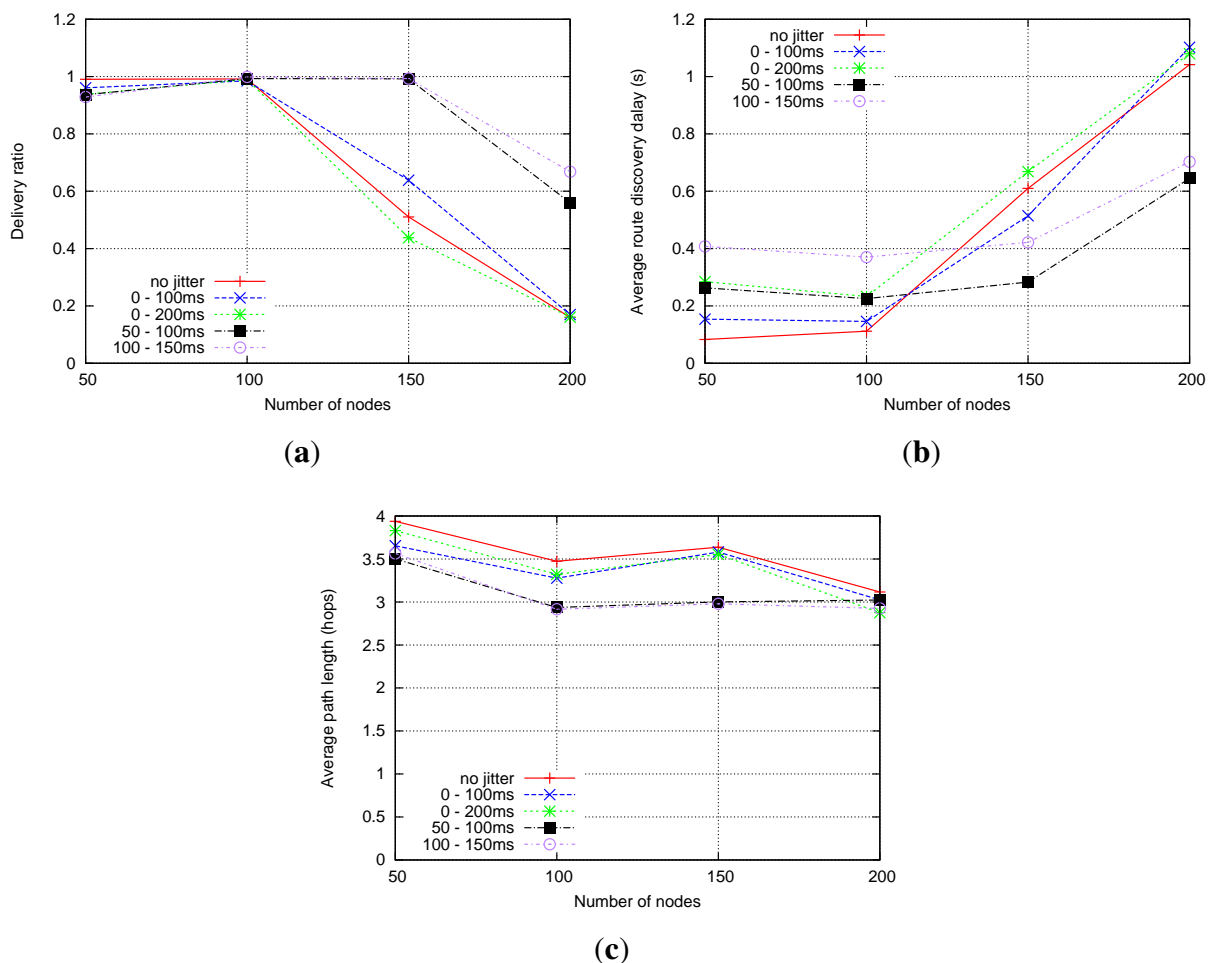
6.2.2. Shortest-Delay RREQ Forwarding

As only the first RREQ is forwarded, the network tends to experience the same overhead under the shortest-delay strategy; therefore, the number of collisions is similar regardless of which type of jitter is used, as shown in Figure 18a. The number of collisions in no-jitter settings, in contrast, is obviously higher. In this situation of similar control traffic overhead, those settings using window jitter distributions have an expectedly longer average route discovery delay, as depicted in Figure 18b.

Figure 18. Simulation results of shortest-delay RREQ forwarding. (**a**) The number of collisions; (**b**) The average route discovery delay; (**c**) Average path length.

(**a**)

(**b**)

(**c**)
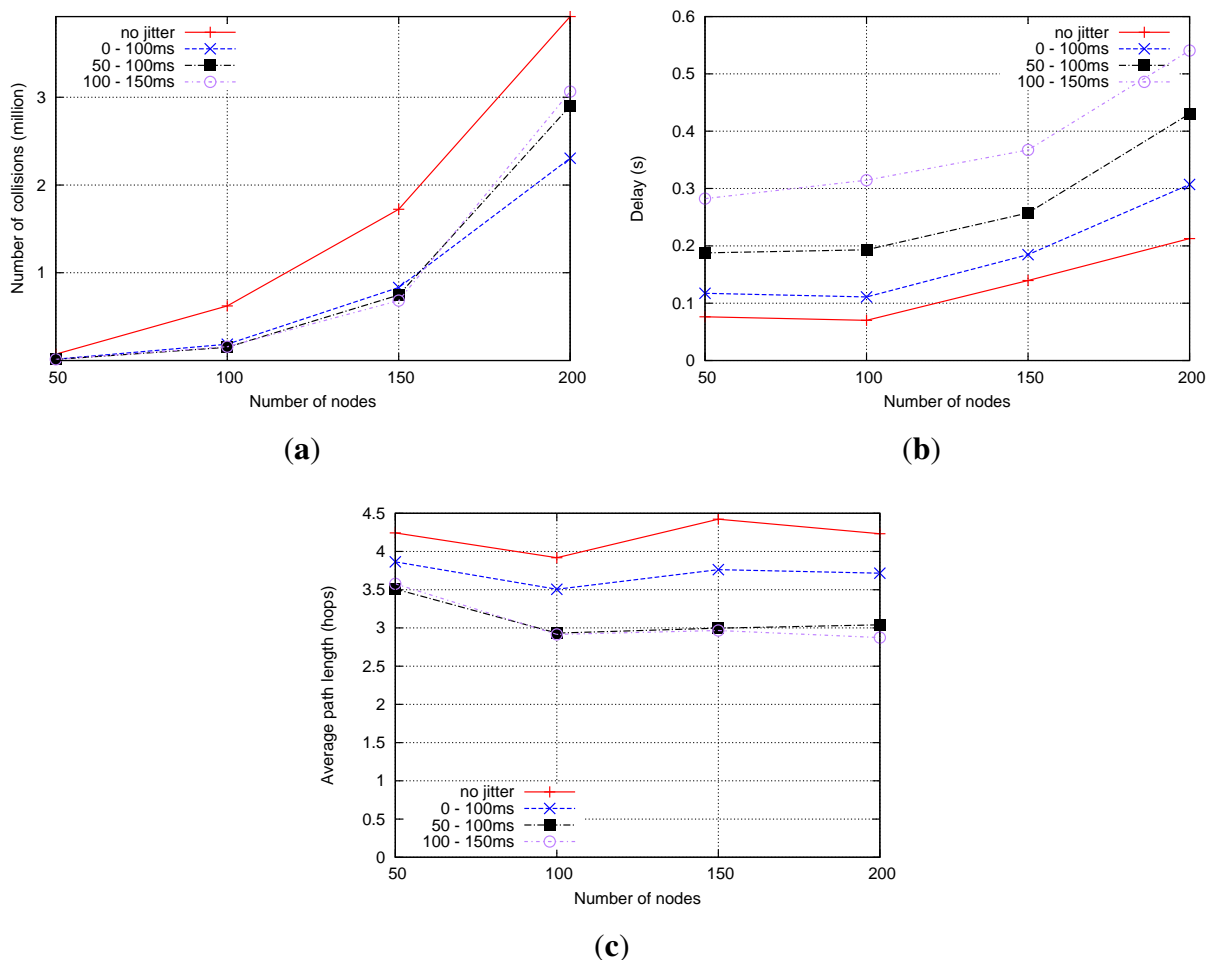
However, because the intermediate routers simply forward the RREQ that arrives first and all of the following RREQs are ignored, the no jitter and "standard" jitter settings got sub-optimal routes. In the meantime, with "window jitter", a much shorter routes can be explored, as illustrated in Figure 18c. This is more interesting for less time-critical, but power-constrained, networks (such as sensor networks).

7. Discussion

Reactive routing protocols rely on flooding, as part of their route discovery process. In wireless networks, flooding incurs the risk of on-the-media collisions and resulting data losses, with the use of

jitter on packet (re-)transmissions being the remedy commonly employed for alleviating these losses. The performance of the flooding operation has, as this paper has presented, a direct impact on the quality of the routes discovered. A similar conclusion was found in [25], in which not jitter, but different flooding algorithms were compared for their impact on route quality when used as part of the route discovery process of a reactive routing protocol.

For the purpose of route discovery, the performance of a flooding operation can be evaluated through three main criteria: (1) the resulting quality (path lengths) of discovered routes; (2) the delay incurred when discovering a new route; and (3) the number of collisions caused by route request flooding. Collisions and route quality have a direct impact on the network load: packet collisions may lead to retransmissions, longer paths imply that more transmissions happen than would otherwise be required between a source and a destination. Alas, with respect to jitter, it is not possible to optimize a flooding operation according to all criteria: an infinitely large jitter interval might, for example, eliminate collisions, but would incur unacceptable delays before routes were established. This section summarizes the trade-offs achieved by the different mechanisms proposed and evaluated in this paper.

It is well established that pure flooding (*i.e.*, without jitter) in wireless multi-hop networks entails an unacceptable level of packet collisions. The use of "standard jitter" (*i.e.*, uniform jitter according to RFC 5148 [9] distributed within $[0, J_m]$) reduces the amount of flooding collisions, at the expense of: (1) slowing down the flooding process; and (2) causing the "delay inversion" effect, as described in Section 3. According to the analysis of Section 4, delay inversion does not depend on the length of jittering time interval (J_m) and becomes more relevant in long paths; that is, in big-diameter networks.

Window jitter introduces a minimum non-zero pre-forwarding delay (αJ_m). This dramatically reduces the probability that delay inversion occurs, and thereby an improvement of the quality of discovered routes (in the shortest-delay mode) or a reduction in control overhead and packet collisions (in the shortest-path mode). It also increases the number of collisions in the first hop of flooding of the flooding process (*i.e.*, when an RREQ is retransmitted by routers one hop away from the source), as jitter selection becomes more deterministic. Further evaluation of the performance of RREQ flooding, with "standard" jitter and window jitter, via graph and network simulations (Sections 5 and 6), provide empirical evidence of the route quality improvement with window jitter and shows that the number of collisions along the explored paths is also reduced with this mechanism. Additional collisions in the first hop are thus largely compensated for by the reduction in the number of transmissions farther away; an effect that becomes relevant as the network density increases. Figure 19 provides a qualitative representation of the different trade-offs achieved by the three main variants (no jitter, uniform and window jitter).

Window jitter outperforms standard jitter in hop count networks, but performs poorly when link quality values are heterogeneous. In scenarios in which link quality information is available across the network (for instance, via link quality estimation), adaptive jitter presents clear advantages with respect to the two static configurations (standard and window jitter). In adaptive jitter, pre-forwarding delays depend on the quality of the link through which the packet was received. Not surprisingly, the performance is then better than in static jitter configurations, both in terms of control overhead and route quality. Under the shortest-path mode, route quality is similar for all settings, but adaptive jitter reduces the number of RREQs and RREPs up to 30%. There is still a trade-off between route quality and

flooding delay, meaning that adaptive jitter entails additional delays; such a delay becomes negligible as the network density grows.

Figure 19. Comparative diagram between no jitter (circle), uniform or standard jitter (square) and window jitter (triangle).
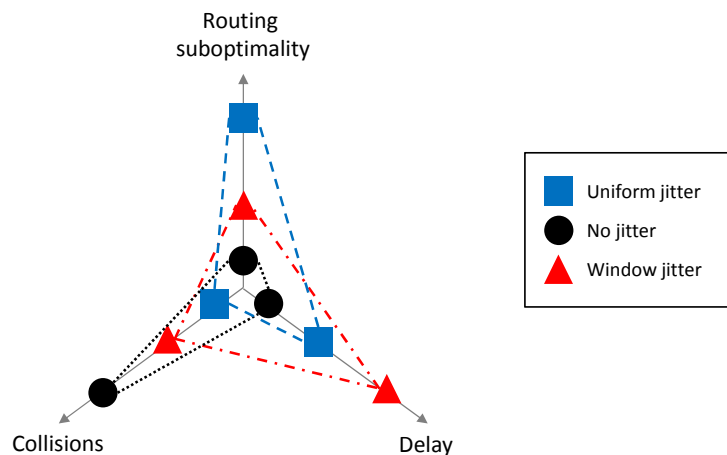
8. Conclusions

Jitter is used for message flooding in wireless multi-hop networks, in order to avoid data losses. It has been shown to be an efficient remedy in proactive routing protocols, such as OLSR [10], and has been recommended for general use by the IETF in RFC 5148 [9].

However, when applied to the route discovery process in reactive routing protocols, using the jitter mechanism prescribed by RFC 5148 [9] has several undesirable side-effects. Observed experimentally, this paper identifies and quantifies analytically the "delay inversion", which may result in unnecessary control traffic overhead being generated and suboptimal paths to the destination ultimately produced.

Not using jitter is not an option: route discovery relies on the flooding of RREQs, and with intermediate routers immediately retransmitting a received RREQ, collisions will still occur and (RREQ) packets will be lost. Therefore, this paper explores ways to employ jitter on RREQ flooding for reactive routing protocols, while minimizing the impact of delay inversion. In particular, window jitter and adaptive jitter, are compared to "standard" jitter (*i.e.*, jitter according to [9]); and, of course, to naive flooding (*i.e.*, immediate retransmission without jittering). This is not only in terms of incurred delay, but also in terms of avoided collisions, control overhead and quality of discovered routes. Comparisons have been performed by way of both probabilistic analysis and experimental analysis through graph and network (NS2)-based simulations.

The proposed "adaptive jitter" is a simple modification to "standard" jitter, yet despite this simplicity, 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. In its most general version, it simply consists of sending flooded messages sooner across better links, which entails a higher probability of the route discovery process resulting in optimal routes. This paper also details the impact of jitter parameters and distribution, on flooding and route discovery performance. Depending on the characteristics of the network, the information available to the routers and the specific route discovery

mode, different trade-offs between the main involved parameters (control overhead, route optimality, flooding delay) can be achieved with the proposed jitter distributions.

The results, pretended in this paper, indicate that window jitter (and adaptive jitter, if routers can reliably estimate wireless link quality) is particularly beneficial for reducing overhead in dense networks, in which packet collisions are more frequent and, therefore, delay inversion more likely to happen, due to the the diversity of available paths. These jitter mechanisms thus are interesting for large resource constrained networks, such as in battery-powered mesh and sensor networks, where extraneous transmissions and long paths are particularly harmful to the network lifetime. Future work should see to confirm these results, both through additional analysis, more complex network simulations and real-world experiments. In particular, future work should study the impact of router mobility and wireless channel unreliability on the applicability of window and adaptive jitter.

Author Contributions

Jiazi Yi discovered the issue of the delay inversion effect in reactive protocols, proposed the adaptive jitter mechanism to alleviate the impacts and conducted the network simulations. Juan Antonio Cordero built the analytical model of the jitter mechanisms and designed and performed the graph simulations. Thomas Clausen provided important contributions to the study of uniform jitter and reactive routing protocols.

Appendix

A. Proofs

Proposition 1. From Equation (2), we have:

$$
\begin{aligned}
D^{(n,m)} &= T^{(n)} - T^{(m)} = \sum_{i=1}^{n} T_{j_i} - \sum_{j=1}^{m} T_{j_j} = \\
&= \sum_{i=1}^{n} T_{j_i} + \sum_{j=1}^{m} \left(-T_{j_j} \right)
\end{aligned}
$$

In the case of $D_U^{n,m}$, T_j is a random variable uniformly distributed within $[0, J_m]$.

Therefore, $f_{D_U^{n,m}}(t)$ can be computed by using the well-known result for the pdf of the sum, S_n, of n independent uniform random variables $Uniform[0,1]$ [26]:

$$
f_{S_n}(x) = \frac{1}{(n-1)!} \sum_{k=0}^{n} (-1)^k \binom{n}{k} (x-k)_+^{n-1} \tag{A1}
$$

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 $Pr(D_U^{(n,m)} > 0)|_{n<m}$. The expression of $P_U \equiv Pr(D_U^{(n,m)} > 0)$ is computed in Equation (A2):

$$P_U = \frac{1}{(n+m)!} \sum_{k=0}^{n+m} \left[(-1)^k \binom{n+m}{k} \times \right.$$
$$\left. \times \left((n+2m-k)^{n+m} - (m-k)_+^{n+m} \right) \right] \tag{A2}$$

$\square$

Proposition 2.

$$f_D(t) = \left(\bigotimes_{i=1}^{n} f_{T_{j\,i}^*}(t) \right) \otimes \left(\bigotimes_{i=1}^{m} f_{(-T_{j\,i}^*)}(t) \right) \tag{A3}$$

$$\begin{cases} f_{T_{j\,i}^*}(t) = \frac{1}{1-\alpha} p\left(\frac{t-\alpha}{1-\alpha} \right) \\ f_{(-T_{j\,i}^*)}(t) = \frac{1}{1-\alpha} p\left(\frac{-t-\alpha}{1-\alpha} \right) = \frac{1}{1-\alpha} p\left(\frac{t+1}{1-\alpha} \right) \end{cases} \tag{A4}$$

where:

$$p(t) = Heaviside(t) - Heaviside(t-1)$$

Then, Equation (A3) becomes, by using Equation (A4):

$$(1-\alpha)^{n+m-1} \times \frac{1}{(1-\alpha)^{n+m}} \left(\bigotimes_{i=1}^{n+m} p \right) \left(\frac{t-n\alpha+m}{1-\alpha} \right) \tag{A5}$$

Note that $\left(\bigotimes_{i=1}^{n+m} p \right)(x)$ is the pdf of the sum of $n+m$ uniform independent random variables within $[0,1]$. Therefore, by applying Equation (A1), Equation (A5) becomes:

$$\frac{(1-\alpha)^{-1}}{(n+m-1)!} \sum_{k=0}^{n+m} (-1)^k \binom{n+m}{k} \left(\frac{t-n\alpha+m}{1-\alpha} - k \right)_+^{n+m-1} \tag{A6}$$

$\square$

Proposition 3. From the definition of probability, $Pr(D_W > 0) = \int_0^\infty f_{D_W}(x)dx$, which is:

$$P_W = \int_0^\infty \frac{dt \sum_{k=0}^{n+m} (-1)^k \binom{n+m}{k} \left(\frac{t-n\alpha+m}{1-\alpha} - k \right)_+^{n+m-1}}{(1-\alpha)(n+m-1)!} =$$
$$= \frac{\sum_{k=0}^{n+m} (-1)^k \binom{n+m}{k} \int_0^\infty dt \left(\frac{t-n\alpha+m}{1-\alpha} - k \right)_+^{n+m-1}}{(1-\alpha)(n+m-1)!} \tag{A7}$$

The integrand of Equation (A7) is zero when $\frac{t-n\alpha+m}{1-\alpha} > n+m \iff t > n - m\alpha$, and $n - m\alpha > 0$ (see Proposition 2); therefore, Equation (A7) can be rewritten as follows:

$$P_W = \frac{\sum_{k=0}^{n+m} (-1)^k \binom{n+m}{k} \int_0^M dt \left(\frac{t-n\alpha+m}{1-\alpha} - k \right)_+^{n+m-1}}{(1-\alpha)(n+m-1)!} \tag{A8}$$

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:

$$\frac{t - n\alpha + m}{1 - \alpha} - k > 0 \Longrightarrow t > k(1 - \alpha) + n\alpha - m \equiv t_0 \tag{A9}$$

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

$$I_0 \;=\; \int_{\max\{0,t_0\}}^{\max\{0,n-m\alpha\}} dt \left(\frac{t - n\alpha + m}{1 - \alpha} - k\right)^{n+m-1} \tag{A10}$$

t_0 depends on k. It is worth observing that:

- $t_0 < n - m\alpha, \forall k : 0 \leq k \leq n + m$. And $n - m\alpha > 0$ if $\alpha > \frac{n}{m}$.

- $t_0 > 0 \Longleftrightarrow k(1 - \alpha) + n\alpha - m > 0$, 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:

$$I_0 \;=\; \int_{\max\{0,t_0\}}^{\max\{0,n-m\alpha\}} dt \left(\frac{t - n\alpha + m}{1 - \alpha} - k\right)^{n+m-1} \tag{A11}$$

Then, Equation (A8) can be computed as follows:

$$P_W = \begin{cases} 0 & , \alpha > \frac{n}{m} \\ \frac{1}{(n+m)!} \sum_{k=0}^{n+m} \left[(-1)^k \binom{n+m}{k} \times \right. \\ \times \left((n + m - k)^{n+m} - (n + \frac{m}{1-\alpha})^{n+m})\right] & , \frac{m}{n} < \alpha \leq \frac{n}{m} \\ \frac{1}{(n+m)!} \left(\sum_{k=0}^{n+m}(-1)^k \binom{n+m}{k}(n + m - k)^{n+m} \right. \\ \left. - \sum_{k=0}^{k^*}(-1)^k \binom{n+m}{k} \left(\frac{m - n\alpha}{1-\alpha} - k\right)^{n+m}\right) & , \alpha \leq \frac{n}{m}, \frac{m}{n} \end{cases} \tag{A12}$$

□

Lemma 1. In the described scenario (see Figure 7) with n neighbors ($n > 1$), there is an auto-collision if the minimum inter-arrival time is smaller than the transmission time for a packet, $L(p)$. Since inter-arrival times $\delta_i = |T_{j_i} - T_{j_{i+1}}|$ are independent, according to Poisson's hypothesis, we have:

$$\begin{aligned} Pr(\text{auto-collision}) &= P(\min_{0 < i \leq n} \{|T_{j_i} - T_{j_{i+1}}|\} < L) = \\ &= \prod_{0 < i \leq n} Pr(\{|T_{j_i} - T_{j_{i+1}}|\} < L) = \\ &= Pr(\{|T_{j_i} - T_{j_{i+1}}|\} < L)^n \end{aligned}$$

The distribution of δ_i is exponential with rate λ; therefore:

$$\begin{aligned} Pr(\text{auto-collision}) &= Pr(\{|T_{j_i} - T_{j_{i+1}}|\} < L)^n = \\ &= \left(\int_0^L \lambda e^{-\lambda x} dx\right)^n = \\ &= \left(e^{-\lambda L} - 1\right)^n \end{aligned} \tag{A13}$$

Expression ([A13](#)) becomes, for uniform jitter ($\lambda_{\text{uniform}} = \frac{n}{J_m}$):

$$Pr(\text{auto-collision})_{\text{uniform}} \quad = \quad \left(e^{-\frac{n}{J_m}L} - 1 \right)^n$$

Additionally, for window jitter, with parameter α ($\lambda_{\text{window}} = \frac{n}{(1-\alpha)J_m}$):

$$Pr(\text{auto-collision})_{\text{uniform}} \quad = \quad \left(e^{-\frac{n}{(1-\alpha)J_m}L} - 1 \right)^n$$

$\square$

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Ni, S.Y.; Tseng, Y.C.; Chen, Y.S.; Sheu, J.P. The Broadcast Storm Problem in a Mobile Ad Hoc Network. In Proceedings of the ACM Annual International Conference on Mobile Computing and Networking (MobiCom'99), Seattle, WA, USA, August 1999.
2. Qayyum, A.; Viennot, L.; Laouiti, A. Multipoint relaying for flooding broadcast messages in mobile wireless networks. In Proceedings of the Hawaii International Conference on System Sciences (HICSS'35), Big Island, HI, USA, 7–10 January 2002.
3. Sasson, Y.; Cavin, D.; Schiper, A. Probabilistic Broadcast for Flooding in Wireless Mobile Ad Hoc Networks. In Proceedings of the IEEE Conference on Wireless Communications and Networking (WCNC 2003), New Orleans, LA, USA, 20 March 2003.
4. Baccelli, E.; Cordero, J.A.; Jacquet, P. Optimization of Critical Data Synchronization via Link Overlay RNG in Mobile Ad Hoc Networks. In Proceedings of the 7th IEEE International Conference Mobile Ad Hoc and Sensor Systems (MASS), San Francisco, CA, USA, 8–12 November 2010.
5. Kleinrock, L.; Tobagi, F.A. Packet switching in radio channels: Part I—carrier sense multiple-access modes and their throughput-delay characteristics. *IEEE Trans. Commun.* **1975**, *23*, 1400–1416 .
6. Karn, P. MACA, A New Channel Access Method for Packet Radio. In Proceedings of the 5th ARRL/CRRL Computer Networking Conference, London, ON, Canada, September 1990; pp. 134–140.
7. Yu, Q.; Chen, J.; Fan, Y.; Shen, X.S.; Sun, Y. Multi-Channel Assignment in Wireless Sensor Networks: A Game Theoretic Approach. In Proceedings of the 29th IEEE International Conference on Computer Communications (INFOCOM 2010), San Diego, CA, USA, 14–19 March 2010.
8. Wu, Y.; Stankovic, J.A.; He, T.; Lin, S. Realistic and Efficient Multi-Channel Communications in Wireless Sensor Networks. In Proceedings of the 27th IEEE International Conference on Computer Communications (INFOCOM 2008), Phoenix, AZ, USA, 13–18 April 2008.
9. Clausen, T.; Dearlove, C.; Adamson, B. Jitter Considerations in MANETs. IETF Request for Comments 5148. February 2008.

10. Clausen, T.; Jacquet, P. Optimized Link State Routing Protocol (OLSR). IETF Request for Comments 3626. October 2003.

11. Clausen, T.; Dearlove, C.; Jacquet, P. Optimized Link State Routing Protocol Version 2 (OLSRv2). IETF Request for Comments 7181. April 2014.

12. Baccelli, E.; Jacquet, P.; Nguyen, D.; Clausen, T. OSPF Multipoint Relay (MPR) Extension for Ad Hoc Networks. IETF Request for Comments 5449. February 2009.

13. Cordero, J.A.; Clausen, T.; Baccelli, E. Mpr+sp: Towards a unified mpr-based manet extens ion for ospf. In Proceedings of the IEEE 2011 44th Hawaii International Conference on System Sciences (HICSS), Kauai, HI, USA, 4 January 2011; pp. 1–10.

14. Friedman, R.; Hay, D.; Kliot, G. *Jittering Broadcast Transmissions in MANETs: Quantification and Implementation Strategies*; Technical Report; Department of Computer Science, Israel Institute of Technology, Haifa, Israel, 2009.

15. Cordero, J.A.; Jacquet, P.; Baccelli, E. Impact of Jitter-based Techniques on Flooding over Wireless Ad hoc Networks: Model and Analysis. In Proceedings of the 31st IEEE International Conference on Computer Communications (INFOCOM 2012), Orlando, FL, USA, 25–30 March 2012.

16. Cordero, J.A.; Yi, J.; Clausen, T. Optimization of Jitter Configuration for Reactive Route Discovery in Wireless Mesh Networks. In Proceedings of the 11th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt 2013), Tsakuba, Japan, 13–17 May 2013.

17. Yi, J.; Cordero, J.A.; Clausen, T. Jitter Considerations in On-Demand Route Discovery for Mobile Ad Hoc Networks. In Proceedings of the 16th International Conference on Network-Based Information Systems (NBiS 2013), Gwangju, Korea, 4 September 2013.

18. Perkins, C.; Belding-Royer, E.; Das, S. Ad hoc On-Demand Distance Vector (AODV) Routing. IETF Request for Comments 3561. July 2003.

19. Johnson, D.; Hu, Y.; Maltz, D. The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4. IETF Request for Comments 4728. February 2007.

20. Clausen, T.; de Verdiere, A.C.; Yi, J.; Niktash, A.; Igarashi, Y.; Satoh, H.; Herberg, U. The LLN On-demand Ad Hoc Distance-Vector Routing Protocol—Next Generation. IETF Internet Draft (work in process) *draft-clausen-lln-loadng*. 3 July 2014.

21. International Telecommunication Union, ITU-T G.9903: Narrow-band orthogonal frequency division multiplexing power line communication transceivers for G3-PLC networks. Telecommunication Standardization Sector of ITU. May 2013.

22. Kim, K.; Park, S.D.; Montenegro, G.; Yoo, S.; Kushalnagar, N. 6LoWPAN Ad Hoc On-Demand Distance Vector Routing. IETF Internet Draft. June 2007.

23. Karmeshu. *Entropy Measures, Maximum Entropy Principle and Emerging Applications*, 1st ed.; Springer: Berlin, Germany, 2003.

24. De Couto, D.S.J.; Aguayo, D.; Bicket, J.; Morris, R. 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, CA, USA, 14 September 2003; p. 8.

25. Clausen, T.; Jacquet, P.; Viennot, L. Optimizing Route Length in Reactive Protocols for Ad Hoc Networks. In Proceedings of The First Annual Mediterranean Ad Hoc. Networking Workshop (Med-hoc-Net 2002), Sardegna, Italy, September 2002

26. Feller, W. *An Introduction to Probability Theory and its Applications*, 3rd ed.; John Wiley & Sons: Hoboken, NJ, USA, 2008.