Influence of beta and source packet rate on electromagnetic nanocommunications

Farah Hoteit*, Eugen Dedu*, Winston K.G. Seah† and Dominique Dhoutaut*

*FEMTO-ST Institute, Univ. Bourgogne Franche-Comté, CNRS
Montbéliard, France
Email: firstName.lastName@univ-fcomte.fr

†School of Engineering and Computer Science, Victoria University of Wellington
Wellington, New Zealand
Email: winston.seah@ecs.vuw.ac.nz

***Abstract*—Nanotechnology permits the manipulation of materials at nanoscale. Contrarily to traditional ad hoc wireless networks, electromagnetic nanonetworks are massively dense networks of highly-constrained nanodevices, connected by a ultra high-capacity terahertz channel (in the order of Tb/s). Like in traditional networks, congestion and collisions can appear in nanonetworks. However, they have a different meaning, and the parameters influencing them are also different. A parameter specific to nanonetworks is beta (β), which expresses the ratio of time between two consecutive bits and the bit length, and is sometimes called the symbol rate. No paper in the literature has evaluated this parameter, neither congestion and collisions in a dense network. Therefore, in this paper we study the influence of this parameter together with source packet rate to congestion and collisions in the context of a multi-flow nanonetwork. We conclude that lower source rates and lower β values result in less congestion and thus a good packet delivery to the destination.**

***Index Terms*—Congestion, Routing, Nanonetwork, Dense network, Scalability**

I. Introduction

Nanotechnology offers the engineering domain a novel manipulation of materials at the nanoscale, which provides new functionalities and presents new behaviors of materials. Nanomaterials open the door for a new paradigms of communication, mainly electromagnetic, and molecular nanocommunications. This paper focuses on electromagnetic nanonetworks, where integrated nanosensor devices connect in the terahertz band [1].

Electromagnetic nanonetworks are envisioned to revolutionize many application domains, which include nanomedecine and software-defined metamaterials. For example, nanomedecine proposes to introduce nanodevices in the human body for health monitoring systems and Drug Delivery Systems (DDS) [2]. In addition, in Software Defined Metamaterials (SDMs), nanomachines are proposed to alter the geometry of metamaterials and tune their electromagnetic behavior through simple user commands [3].

Due to their tiny size, nanodevices are low-power passive devices and are limited in their storage, sensing, processing and actuation tasks. The dense deployment of nanodevices in a nanonetwork expands their potential (thousands of nodes in the network). Nanonetworks are massively dense networks where each node can have hundreds or thousands of neighbors (high network size and node density). Applications that require such networks are software-defined metamaterials and in-body communication [4].

Graphene nanoantennas are proposed for these devices to communicate in the terahertz band (0.1–10 THz) [1]. The low-power nanonode is only able to reach its neighbors in its transmission range, and hence uses multi-hop communications in order to get to a distant node, through intermediate nodes called *forwarders* that relay the data packets. The nanoscale, the THz band and the dense deployment of nodes call for the design of novel protocols for all the network layers. Unfortunately, recent research has focused on designing routing protocols for nanonetworks, while neglecting the transport layer.

For the physical layer, some modulation schemes have been proposed, especially TS-OOK and its variant RD TS-OOK, as seen in Fig. 1 [4]. TS-OOK (Time Spread On-Off Keying) is a pulse based-modulation, that is appropriate to THz nanonetworks because of its simplicity. Nanodevices do not have enough resources to use carrier signals for transmission, and instead communicate with pulses: bit 1 is transmitted as a 100 femtosecond-long ($= T_p$) pulse with energy, and bit 0 as a silence without energy. The time between pulses is T_s, and the symbol rate is $\beta = T_s/T_p$ [5]. Transmitters and receivers are highly synchronised. The problem with TS-OOK is that selecting an optimal β value is complex. If $\beta = 1$, all the symbols of a nanodevice are transmitted in burst and only one nanodevice can access the channel at a time. When we increase β, multiple nanodevices can access the channel simultaneously and the throughput of each of them is thus reduced [6].

Pulses (bits 1) have a short duration of T_p=100 fs to reduce the probability of collisions [7]. Using a big time between pulses T_s allows receivers to decode each T_s instead of sensing the channel continuously. A big T_s also permits for multiple nanodevices to transmit simultaneously over the THz channel, and for receiver to receive packets in parallel, and that is different from the sequential nature of receptions at the

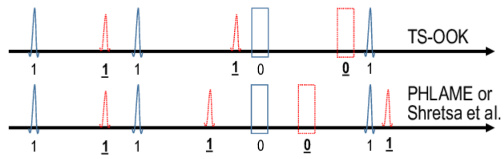

Fig. 1. Comparison between TS-OOK and RD TS-OOK modulations, source: [4], user1 in blue and user2 in red.

macroscale communications. When received, bits from different transmitters may overlap in time and collide. Contrarily to macroscale communications, in nanonetworks catastrophic collisions only occur when the a bit 0 is replaced by a bit 1, and hence some data packets are altered along the route [8].

To minimize collisions [7], RD TS-OOK (Rate Division TS-OOK) makes each transmitting node select its own β randomly from a pool of co-prime rate codes, instead of using the same rate for all nodes as in TS-OOK. β value is included in the transmitting Data Symbol Rate (DSR) field in the Transmission Request (TR) packet generated by the transmitter during the handshaking process in the MAC protocol PHLAME [7]. Another modulation scheme is SRH-TSOOK (Symbol Rate Hopping TSOOK), which makes transmitters change their β after a constant number of symbols that correspond to the transmission of a MAC frame, in order to guarantee uniform use of symbol rates and same transmission conditions for all active communications [9].

Congestion in THz nanonetworks is worth studying since it becomes severe in ultra-dense networks. As seen later, congestion (large number of ignored packet events) in Table II causes packet loss in Table III. In nanonetworks, the low-power nanonodes are unable to consume the wide THz bandwidth as they send short pulses of T_p=100 fs, hence congestion does not arise from the large THz shared channel. Instead, the congestion arises from the *limited buffers of nodes*, and that it is determined at the node level by measuring the buffer level: when the number of packets being received by a node exceeds its capacity, the node is congested [10]. In other words, congestion is the buffer overflow and collision is packets with bits that have been altered. The main source of congestion and collision is the high number of communicated packets, which goes back to the high number of forwarders. Forwarders may be condensed in the same flow, as in inefficient routing protocols. They also may be at the intersection of many flows, which form a congestion zone.

The contribution of this paper is to draw the reader's attention to carefully select appropriate values for β (and thus the modulation scheme) and for the source packet rate, as they affect the congestion and collisions, in order to achieve a good delivery ratio with less costs.

In the rest of this paper, Section II presents the related work. Section III selects the simulation software and describes the scenario. Section IV analyses the results of the simulations. Finally, Section V concludes the paper and gives insights for future work.

II. Related work

We first present works on congestion control in nanonetworks, then in the broader field of wireless sensor networks, and finally in classical IP networks.

In general, congestion control mechanisms can be divided into two categories: traffic control by limiting the transmissions (decreasing source rate or congested nodes rate etc.), and resource control by turning-on network resources (alternative routes, duty cycling etc.) [11].

The transport layer of THz *nanonetworks* remains largely unexplored. We found only two articles on congestion control in electromagnetic nanonetworks. One is congestion control by deviation routing, where data packets deviate from the original route to avoid the congestion zone [10]. The other is an energy-efficient transport layer protocol for electromagnetic and molecular body area nanonetwork, where the sender upon receiving a "halt" signal stops the packet transmission for a predefined time [12]; this scheme uses routing tables in a low density scenario (up to 100 nodes), so it is infeasible in ultra-dense scenarios.

Nanonetworks, on the other hand, share similar aspects with *wireless sensor networks* (WSNs). In a survey on wireless sensor networks [11], some protocols use the source packet rate for congestion control: FUSION combines multiple congestion control techniques and one is source limiting where the hop-by-hop back-pressure reaches the source to decrease its rate [13]. This scheme is costly in ultra-dense nanonetworks, as it generates a new traffic and consumes the energy and memory resources of nanodevices. The Congestion Detection and Avoidance in Sensor Networks (CODA) is a protocol where the source waits for constant acknowledgements from the sink using "regulate bit" in event packets [14]. The dense deployment of nanonodes makes CODA costly as well (high network traffic). In the context of nanonetworks, the relationship between the source rate and congestion is not studied. In this paper, we do not propose a dynamic source packet rate congestion control, but rather study the influence of source rate on congestion in the context of THz nanonetworks. It is also important to recall that the congestion in nanonetworks is caused by the limitations of a nanodevice and the dense deployment of nodes, and not from the (THz very wide) band channel. Also, in a ultra-dense nanonetwork, a nanonode cannot maintain full neighborhood or network knowledge, and the location information may not be available given that embedding hardware modules (such as GPS and RSSI) is complex at the nanoscale. This is different from macroscale wireless networks and this makes traditional congestion control mechanisms infeasible in THz nanonetworks.

Finally, in *IP networks*, when packets are lost, TCP controls the congestion by limiting the traffic at the source. ECN, that extends TCP/IP, makes intermediate routers mark packets in a pre-congestion phase, to also decrease the source traffic [15].

Pulse-based modulations in the THz band use the symbol rate β. We propose to adjust the β value along with

TABLE I
SIMULATION PARAMETERS.

Parameter	Value
Size of simulated area	6 mm * 6 mm
Number of nodes	20 000
Communication range	900 μm
Data packet size	1000 bit
Number of flows	5
Number of packets per flow	100
Routing protocol	SLR backoff
Communication range for SLR addressing phase	250 μm
Backoff redundancy	20
MCR	3
Pulse duration T_p	100 fs
MaxBitError	0

source packet rate to control congestion and collisions in THz nanonetworks.

To the best of our knowledge, this is the first paper that addresses the relationship between β and congestion.

III. SIMULATION ENVIRONMENT

A. Available simulation software

An integrated nanosensor device is still under development, therefore experimental validation cannot be done yet, and software simulations are the current option for validation.

In a comparison of available Internet of Nano-Things simulators [16], several simulators are presented. Nano-Sim [17] and TeraSim [18] are heavy and can only simulate networks of up to to around one thousand nodes. Vouivre [19] can simulate tens of thousands of nodes in the nanonetwork, but it does not study the effect of the packet payload and bits. None of the three have any support for visualization.

On the contrary, BitSimulator [8] is mentioned as the one providing the highest scalability (thousands of nodes). It also implements advanced routing protocols, and features a visualization tool VisualTracer for node activity in the network. BitSimulator uses a discrete event model. Therefore, we select BitSimulator as the most appropriate option for this study. It is free software[1].

B. Scenario

We assume that nanonodes are homogeneous: they all radiate the same power and share the same characteristics of processing power and energy. Another assumption is that nanoantennas are omni-directional.

Table I presents the simulation parameters that we selected for the scenario. The scenario describes a 2D nanonetwork of 20 000 nodes distributed in a square area of 36 mm^2, over 3 horizontal equal bands of different densities: 10 000, 6000 and 4000 nodes per band, as seen in Fig. 2. This is an ultra-dense network that may be required in application domains such as intra-body communications and software-defined metamaterials.

The communication range is 900 μm, and hence the multi-hop communications are done over 6 mm/900 μm $\approx$ 6.6 hops

[1] Available at http://eugen.dedu.free.fr/bitsimulator

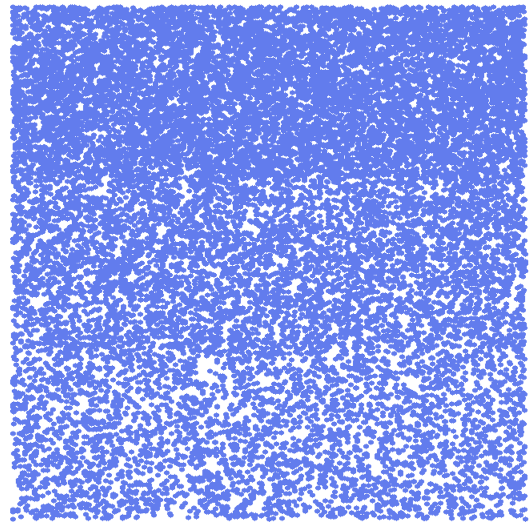

Fig. 2. Nodes distribution in the network (without routing).

in each dimension. Nodes use the normal shadowing propagation model: considering the distance from the transmitter as d, and the communication range as CR ($d < CR$), nodes have a 100% reception rate at a distance in the range of $[0, d]$, a decreasing reception rate from 100% to 0% at a distance in the range of $]d, CR]$, and stop receiving packets at a distance further than CR.

5 CBR (Constant bitrate) flows (source-destination pairs) exist in the scenario, shown in blue in Fig. 3. Each source nodes emits 100 packets (of 1000 bits each) to its corresponding destination node, thus avoiding statistical bias. The 100 packets differ in their random backoff time before transmission, that depends on the node's local density. Source nodes start emitting at the same time. Destination nodes are close to each other so they form a destination zone where the 5 flows intersect and cause a congestion zone. We use the best unicast routing scheme, SLR backoff [10], where two routing protocols (SLR and backoff flooding) are merged.

SLR (Stateless Linear-path Routing) is an addressing and multi-hop unicast (or merely zone cast) routing scheme for electromagnetic nanonetworks. During the addressing phase, anchor nodes generate one SLR beacon each, and nodes set their coordinates as hop counts from these anchors. During the routing phase, source nodes send data packets to the intended destination through forwarders, in a multi-hop transmission. Forwarders are nodes whose coordinates fulfill the source-destination line equation [20].

Backoff flooding [21] is a flooding scheme which uses very few forwarders in a multi-hop communication. The source node intends to broadcast data packets to the whole network through forwarders. To desynchronise node forwarding in ultra-dense networks and reduce collisions, nodes choose a random backoff before forwarding from a very large dynamic time window. Only the nodes having received less than N copies of the data packet in this window forward the packet. The time window is proportional to the node density, given by

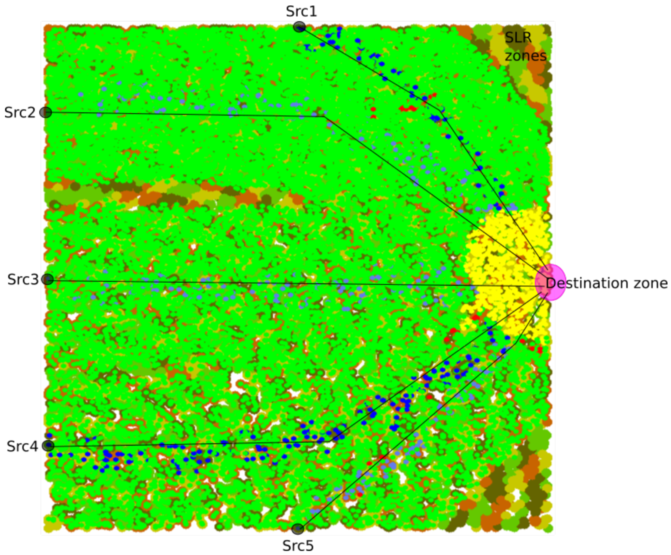

Fig. 3. Scenario with β=1000 and source inter-packet interval=10^5 ns: 5 flows intersecting at destination zone causing a congestion zone (right). Only first packet of each flow is shown. Forwarders are in blue, receivers in green, nodes with ignore in yellow, and nodes with collisions in red. SLR zones in the background.

an ultra-dense network density estimator, such as DEDeN [22].

In the combined SLR backoff protocol, forwarding nodes are on the path between the source and destination (as in SLR) and have also received less than N data copies in the time window (as in backoff flooding). We set the backoff redundancy N (the number of copies above) to 20 to ensure good delivery to the destination(s) and to create congestion. During the SLR addressing phase, nodes only use a fraction of their sending power, and hence a smaller communication range of 250 μm, creating SLR mini-zones, i.e. zones smaller than communication range.

Similarly to other routing protocols, forwarders relay packets they receive the first time and do not forward copies of the same packet, by recording a limited list of the source id-s and the packet sequence numbers of the previously received packets.

Nanonodes are constrained from several points of view: memory, processing power and speed, energy, and various resources. We express all these constraints by a parameter that we call the maximum concurrent receptions (MCR), which fixes the maximum number of packets that a nanodevice can track in parallel (cf. "The receiver can simultaneously track a fixed number of incoming packets, K" [7]). Packets received while the node is already tracking MCR packets are discarded and *ignored*.

We define MaxBitError as the maximum number of altered bits in a collided packet before it is considered damaged and is discarded. The rationale is that error correction codes, such as SBN (Simple Block Nanocode for nanocommunications) [23], can correct up to a given number of bits per packet. Also, we consider that not all collisions are destructive: only bits 0 colliding with bits 1 are altered (cf. "there are no collisions between silences, and collisions between pulses and silences are only harmful from the silence perspective" [7]).

To allow the reproducibility of the simulation results, we provide a separate web site that explains how to regenerate the obtained data[2].

IV. Effect of beta and packet rate on nanocommunications

In order to study the effects of the symbol rate β and source packet rate (or alternatively source inter-packet interval) on communication, three metrics are used: the number of ignore events, the number of collision events, and the percentage of delivery at each destination node. The source inter-packet interval sets the time between the sending of the first bit of two consecutive packets and can be modified only if the application permits it.

We change β for all nodes in the routing phase only (for data packets only). This is because modifying β in the SLR and DEDeN initialization phases results in different SLR zones and local densities respectively, and causes a faulty comparison, as multiple variables are being changed simultaneously; in these two initialization phases we set β to 1000, a value given in the literature ("the ratio between the time between pulses and the pulse duration is kept constant and equal to $\beta = 1000$" [5]).

The packet generated by the source crosses multiple hops and is re-transmitted by many forwarders to get to the destination. This makes nodes receive the same packet multiple times (the first received packet and its copies). A large number of nodes (along the path between the source-destination pairs) receive the data packets and their copies, and thus a huge number of packets are processed, some of them may be ignored, collided or received successfully. Each event generated in the simulations corresponds to one data copy of one node, which further explains the high number of events in Table II.

A. *Effect of symbol rate beta and source packet rate on ignore and collision*

Table II shows how different combinations of β and source packet rate affect the number of ignore and collision events. For instance, the same source inter-packet interval of 100 000 ns for node β values of 100, 1000 and 10 000 causes 74 070, 88 138 and 3 980 873 ignore events, and 418 553, 45 139 and 19 718 collisions, respectively. In other words, the lower the β value, the lower the number of ignores and the higher the number of collisions. This is also seen in Fig. 4, where β=100 exhibits fewer ignores at the congestion zone (yellow) and more collisions (red) than β=1000, in Fig. 3. The explanation of this result is that when reducing β, the time between pulses T_s is reduced too, and symbols are processed faster and thus packets get out faster from buffers (lower ignores). Smaller values of β also mean that fewer simultaneous transmitters share the channel, but more collisions occur at the receiver, because symbols of multiple packets are likely to overlap in time.

[2]http://eugen.dedu.free.fr/bitsimulator/icpads22

TABLE II
NUMBER OF IGNORE AND COLLISION EVENTS FOR 5 FLOWS OF 100 PACKETS EACH IN A NETWORK OF 20 000 NODES.

	Ignore events	Collision events
β=10:		
source inter-packet=10 ns	9 352 176	17 097 957
β=100:		
source inter-packet=10^2 ns	7 334 666	1 687 428
source inter-packet=10^3 ns	4 339 230	2 023 881
source inter-packet=10^4 ns	105 023	471 747
source inter-packet=10^5 ns	74 070	418 553
source inter-packet=10^6 ns	74 070	418 553
β=1000:		
source inter-packet=10^3 ns	6 658 649	161 758
source inter-packet=10^4 ns	3 819 608	198 475
source inter-packet=10^5 ns	88 138	45 139
source inter-packet=10^6 ns	82 603	41 179
β=10 000:		
source inter-packet=10^3 ns	9 277 891	11 843
source inter-packet=10^4 ns	7 109 044	14 494
source inter-packet=10^5 ns	3 980 873	19 718
source inter-packet=10^6 ns	93 691	4 544

TABLE III
PERCENTAGE OF PACKET DELIVERY BY EACH DESTINATION NODE FOR 5 FLOWS OF 100 PACKETS EACH IN A NETWORK OF 20 000 NODES.

Destination	1	2	3	4	5
β=10:					
source inter-pkt=10 ns	13%	18%	18%	13%	15%
β=100:					
source inter-pkt=10^2 ns	21%	21%	22%	13%	18%
source inter-pkt=10^3 ns	82%	88%	84%	76%	75%
source inter-pkt=10^4 ns	100%	100%	100%	100%	100%
source inter-pkt=10^5 ns	100%	100%	100%	100%	100%
source inter-pkt=10^6 ns	100%	100%	100%	100%	100%
β=1000:					
source inter-pkt=10^3 ns	17%	21%	20%	13%	15%
source inter-pkt=10^4 ns	77%	82%	88%	76%	79%
source inter-pkt=10^5 ns	100%	100%	100%	100%	100%
source inter-pkt=10^6 ns	100%	100%	100%	100%	100%
β=10 000:					
source inter-pkt=10^3 ns	5%	4%	4%	3%	5%
source inter-pkt=10^4 ns	17%	21%	19%	19%	16%
source inter-pkt=10^5 ns	78%	88%	90%	74%	77%
source inter-pkt=10^6 ns	100%	100%	100%	100%	100%

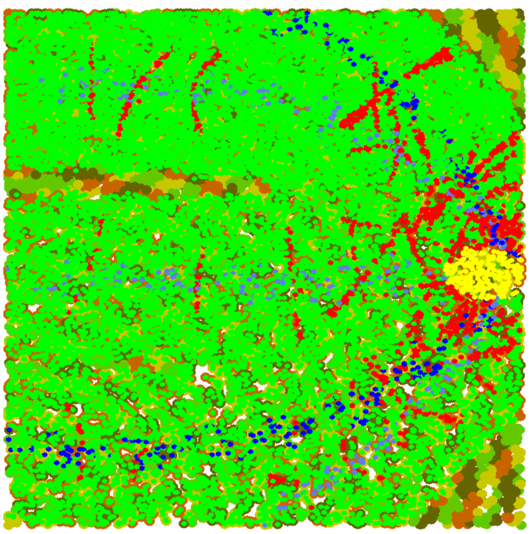

Fig. 4. Scenario used, with β=100 and source inter-packet interval=10^5 ns. Only first packet of each flow is shown.

In the same Table II, for a given β value, 1000 for example, for different source inter-packet intervals, of 1000, 10 000, 100 000 and 1 000 000, the number of ignores are respectively: 6 658 649, 3 819 608, 88 138 and 82 603, and collisions are 161 758, 198 475, 45 139 and 41 179. This means that lower values for the source inter-packet interval induce higher ignore and higher collisions, as expected because of the high number of packets present in the network, which fills buffers faster and leads to higher chances of collisions as bits from many packets are likely to collide.

To sum up, lower β values increase the chances of collisions but reduce the congestion (buffer overflow), and vice-versa; lower source inter-packet intervals stimulate congestion and collisions, and vice-versa.

B. Effect of beta and source packet rate on packet delivery

Packet delivery is computed as the percentage of the number of unique packets received by the destination. In particular, 100% (full) delivery is achieved when the destination receives the 100 packets from its corresponding source.

Table III shows the delivery percentage at each of the five destination nodes. When comparing with Table II, we can conclude that the symbol rate/source packet rate pair that results in high levels of ignores (in the scale of million events) makes the destinations lose some packets. Packet collision is more tolerated than packet ignore, as certain packets can be altered along the route but others arrive successfully to the destinations.

In other words, the high congestion induced by non-optimal selected values for β and source rate prevents packets to reach the destination(s).

C. Effect of dynamic beta on ignore, collision and packet delivery

In TS-OOK modulation, all the nodes use the same β value, of 1000 in our simulations; given that the distance between two consecutive bits is the same, packets which collide in one pulse collide in all pulses [7], [24]. In contrast, in RD TS-OOK, a variant of TS-OOK, transmitters select their own β, randomly from co-prime numbers, aiming to minimize the chances of collisions per packet [7]. We have implemented this part of RD TS-OOK in BitSimulator and assigned to each node a random number from three choices right before each transmission, i.e. a dynamic β. The three choices are taken from the literature: 1009, 1013 and 1019 (“the RD TS-OOK symbol rates are randomly chosen by each node from a pool of pairwise coprime rate codes in the order of 1000 (e.g., 1009, 1013, 1019)” [7]). We chose values close to 1000 also because we want to test the dynamicity, and not the different value of β. As a corollary, given that β values are bound to

TABLE IV
COMPARISON RESULTS BETWEEN TS-OOK AND RD TS-OOK FOR 5 FLOWS OF 100 PACKETS EACH IN A NETWORK OF 20 000 NODES.

	Ignore	Collision	Delivery to dest
source inter-packet=10^5 ns			
TS-OOK β=1000:			
MaxBitError=0	88 138	45 139	100%
MaxBitError=5	83 756	46 014	100%
RD TS-OOK (dynamic β):			
MaxBitError=0	102 820	1 756 377	100%
MaxBitError=5	99 909	30 610	100%

nodes, this means that a packet is transmitted with *different* β as it is routed from node to node.

Table IV presents the influence of dynamic β (TS-OOK and RD TS-OOK) on ignore, collision and packet delivery.

For the same source inter-packet interval of 100 000, β in the order of 1000 (fixed for TS-OOK: 1000, and random for RD TS-OOK: 1009, 1013, 1019) ensures 100% delivery to the destination(s) and give similar values for the ignore events. Here we can also see the relationship between the ignores and the delivery, as low levels of ignore ensure good percentage of delivery.

For the collisions, RD TS-OOK is designed to reduce the collisions by desynchronising the transmitters. However, Table IV shows that the collisions in RD TS-OOK (1 756 377) are notably higher than TS-OOK (45 139), when no bit error is tolerated (MaxBitError=0). To confirm that, compared to TS-OOK, RD TS-OOK causes fewer error bits per packet but more collided packets, we make nodes accept a few error bits. Therefore, when an error correction code of up to 5 bits (MaxBitError=5) is used, RD TS-OOK outperforms TS-OOK: 30 610 collision events (for packets with more than 5 bit errors) for RD TS-OOK compared to 46 014 collisions for TS-OOK.

In brief, RD TS-OOK and TS-OOK give similar numbers of ignores and collisions, as their β are close to each other (close to 1000), no matter if they are dynamic or not. However, RD TS-OOK causes fewer collisions than TS-OOK (fewer bit errors per packet that can be fixed by error correction codes).

V. CONCLUSION AND FUTURE WORK

Nanotechnology could re-engineer the world. Electromagnetic nanonetworks connect tiny nanodevices with expected huge potential. Protocols for all the different layers are being developed, but the transport layer remains the least explored [4] and thus new congestion control protocols are required.

This paper analyses the effects of fixed and dynamic symbol rate of nodes β along with the source packet rate. The results of this paper show that β (and consequently the modulation scheme) and the source rate affect the congestion and collisions. If a congestion is detected in the nanonetwork, users should verify the β and source rate values. Users are encouraged to choose lower source rate (if the application allows it) to avoid congestion, i.e., the source sends then pauses then sends then pauses etc. They are also encouraged to select a lower β value (if the nanomachine hardware permits it) that causes less congestion (buffer overflow), as it makes a faster processing of symbols and packets, which then results in a good packet delivery to the destination(s), even if there are many collisions on the route.

Future work includes theoretical analysis of the effect of β on nanonetworks. The influence of β and the source rate on the delay can be studied. Future research can also be directed towards designing novel congestion control protocols for nanonetworks, such as dynamic source rate limiting traffic, and dynamic β based on node buffer fill level.

ACKNOWLEDGMENTS

This work has been funded by Pays de Montbéliard Agglomération (France).

REFERENCES

[1] J. M. Jornet and I. F. Akyildiz, "Graphene-based nano-antennas for electromagnetic nanocommunications in the terahertz band," in *4th European Conference on Antennas and Propagation*. Barcelona, Spain: IEEE, Apr. 2010, pp. 1–5.

[2] R. Iovine, V. Loscrì, S. Pizzi, R. Tarparelli, and A. M. Vegni, "Electromagnetic nanonetworks for sensing and drug delivery," in *Modeling, Methodologies and Tools for Molecular and Nano-scale Communications*. Springer, 2017, pp. 473–501.

[3] C. Liaskos, A. Tsioliaridou, A. Pitsillides, I. F. Akyildiz, N. V. Kantartzis, A. X. Lalas, X. Dimitropoulos, S. Ioannidis, M. Kafesaki, and C. Soukoulis, "Design and development of software defined metamaterials for nanonetworks," *IEEE Circuits and Systems Magazine*, vol. 15, no. 4, pp. 12–25, Apr. 2015.

[4] F. Lemic, S. Abadal, W. Tavernier, P. Stroobant, D. Colle, E. Alarcón, J. Marquez-Barja, and J. Famaey, "Survey on terahertz nanocommunication and networking: A top-down perspective," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 6, pp. 1506–1543, Apr. 2021.

[5] J. M. Jornet and I. F. Akyildiz, "Femtosecond-long pulse-based modulation for Terahertz band communication in nanonetworks," *IEEE Transactions on Communications*, vol. 62, no. 5, pp. 1742–1753, May 2014.

[6] L. Aliouat, M. Rahmani, H. Mabed, and J. Bourgeois, "Enhancement and performance analysis of channel access mechanisms in terahertz band," *Nano Communication Networks*, vol. 29, p. 100364, 2021.

[7] J. C. Pujol, J. M. Jornet, and J. S. Pareta, "PHLAME: A physical layer aware MAC protocol for electromagnetic nanonetworks," in *2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. Shanghai, China: IEEE, Apr. 2011, pp. 431–436.

[8] D. Dhoutaut, T. Arrabal, and E. Dedu, "BitSimulator, an electromagnetic nanonetworks simulator," in *5th ACM/IEEE International Conference on Nanoscale Computing and Communication (NanoCom)*. Reykjavik, Iceland: ACM/IEEE, Sep. 2018, pp. 1–6.

[9] H. Mabed, "Enhanced spread in time on-off keying technique for dense terahertz nanonetworks," in *22nd IEEE Symposium on Computers and Communications (ISCC)*. Heraklion, Greece: IEEE, Jul. 2017, pp. 710–716.

[10] T. Arrabal, F. Büther, D. Dhoutaut, and E. Dedu, "Congestion control by deviation routing in nanonetworks," in *6th ACM International Conference on Nanoscale Computing and Communication (NanoCom)*. Dublin, Ireland: ACM/IEEE, Sep. 2019, pp. 1–6.

[11] M. A. Kafi, D. Djenouri, J. Ben-Othman, and N. Badache, "Congestion control protocols in wireless sensor networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1369–1390, Mar. 2014.

[12] M. Alam, N. Nurain, S. Tairin, and A. A. Al Islam, "Energy-efficient transport layer protocol for hybrid communication in body area nanonetworks," in *2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*. Dhaka, Bangladesh: IEEE, Dec. 2017, pp. 674–677.

[13] B. Hull, K. Jamieson, and H. Balakrishnan, "Mitigating congestion in wireless sensor networks," in *Proceedings of the 2nd international conference on Embedded networked sensor systems*. New York, NY, USA: Association for Computing Machinery, 2004, pp. 134–147.

[14] C.-Y. Wan, S. B. Eisenman, and A. T. Campbell, “Energy-efficient congestion detection and avoidance in sensor networks,” *ACM Transactions on Sensor Networks (TOSN)*, vol. 7, no. 4, pp. 1–31, Feb. 2011.

[15] K. Ramakrishnan, S. Floyd, and D. Black, “The addition of explicit congestion notification (ECN) to IP,” Sep. 2001, RFC 3168.

[16] E. Sahin, O. Dagdeviren, and M. A. Akkas, “An evaluation of internet of Nano-Things Simulators,” in *6th International Conference on Computer Science and Engineering (UBMK)*. Ankara, Turkey: IEEE, Sep. 2021, pp. 670–675.

[17] G. Piro, L. A. Grieco, G. Boggia, and P. Camarda, “Nano-Sim: simulating electromagnetic-based nanonetworks in the network simulator 3,” in *6th International ICST Conference on Simulation Tools and Techniques (SimuTools)*. Cannes, France: ACM, Mar. 2013, pp. 203–210.

[18] Z. Hossain, Q. Xia, and J. M. Jornet, “TeraSim: An *ns-3* extension to simulate Terahertz-band communication networks,” *Nano Communication Networks*, vol. 17, pp. 36–44, Sep. 2018.

[19] N. Boillot, D. Dhoutaut, and J. Bourgeois, “Going for large scale with nano-wireless simulations,” in *2nd ACM International Conference on Nanoscale Computing and Communication (NanoCom)*. Boston, MA, USA: ACM, Sep. 2015, pp. 1–2.

[20] A. Tsioliaridou, C. Liaskos, E. Dedu, and S. Ioannidis, “Packet routing in 3D nanonetworks: A lightweight, linear-path scheme,” *Nano Communication Networks*, vol. 12, pp. 63–71, Jun. 2017.

[21] T. Arrabal, D. Dhoutaut, and E. Dedu, “Efficient multi-hop broadcasting in dense nanonetworks,” in *17th IEEE International Symposium on Network Computing and Applications (NCA)*. Cambridge, MA, USA: IEEE, Nov. 2018, pp. 385–393.

[22] ——, “Efficient density estimation algorithm for ultra dense wireless networks,” in *27th International Conference on Computer Communications and Networks (ICCCN)*. Hangzhou, China: IEEE, Jul.-Aug. 2018, pp. 1–9.

[23] M. A. Zainuddin, E. Dedu, and J. Bourgeois, “SBN: Simple block nanocode for nanocommunications,” in *ACM International Conference on Nanoscale Computing and Communication*, ser. 3. New York City, NY, USA: ACM, Sep. 2016, pp. 1–7.

[24] A. P. Shrestha, S.-J. Yoo, H. J. Choi, and K. S. Kwak, “Enhanced rate division multiple access for electromagnetic nanonetworks,” *IEEE Sensors Journal*, vol. 16, no. 19, pp. 7287–7296, 2016.