



- (51) **International Patent Classification:**
H04W 28/06 (2009.01)
- (21) **International Application Number:**
PCT/EP2015/077951
- (22) **International Filing Date:**
27 November 2015 (27.11.2015)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
14196569.9 5 December 2014 (05.12.2014) EP
- (71) **Applicant:** PHILIPS LIGHTING HOLDING B.V.
[NL/NL]; High Tech Campus 45, 5656 AE Eindhoven
(NL).
- (72) **Inventors:** STOLIKJ, Milosh; High Tech Campus 5,
5656AE Eindhoven (NL). LUKKIEN, Johan; High Tech
Campus 5, 5656AE Eindhoven (NL). CUIJPERS, Pieter
- Jan Laurens;** High Tech Campus 5, 5656AE Eindhoven
(NL).
- (74) **Agents:** VERWEIJ, P., D. et al.; Philips Lighting B.V.,
Philips Lighting Intellectual Property, High Tech Campus
5, 5656 AE Eindhoven (NL).
- (81) **Designated States** (*unless otherwise indicated, for every
kind of national protection available*): AE, AG, AL, AM,
AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY,
BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM,
DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT,
HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR,
KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG,
MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM,
PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC,
SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN,
TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.
- (84) **Designated States** (*unless otherwise indicated, for every
kind of regional protection available*): ARIPO (BW, GH,

[Continued on next page]

(54) **Title:** METHOD FOR OPERATING A COMMUNICATION DEVICE.

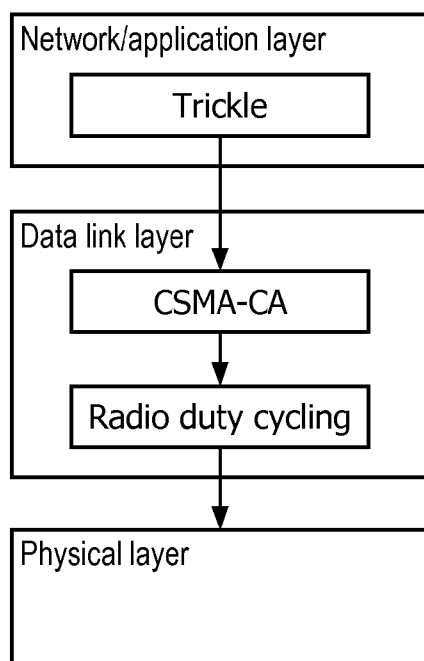


FIG. 1

(57) **Abstract:** The present invention relates to a method for operating a communication device, the method comprising a) the communication device, upon detection of a first event, preparing at the network layer a data frame to be transmitted at a selected transmission time, b) in case a first condition is fulfilled at said transmission time, the communication device buffering the data frame for transmission at a layer lower than the network layer, c) the communication device checking that the first condition is still fulfilled before transmission of the data frame by the lower layer.



GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

Published:

— *with international search report (Art. 21(3))*

METHOD FOR OPERATING A COMMUNICATION DEVICE

FIELD OF THE INVENTION

The present invention relates to a method for operating a communication device, in particular in a network. It also relates to a communication device adapted to carry out such method.

This invention is, for example, relevant for networks such as low-power wireless networks.

BACKGROUND OF THE INVENTION

In a conventional communication device, a communication stack includes a plurality of layers. For example, as shown on Figure 1, the OSI model layer includes different communication having different functions in the handling of data packets and interfacing the applications run on the communication device and the air interface.

Below the application layer, some routing mechanisms are handled by the network layer. In one example, in an ad-hoc network, the network layer may handle the Trickle algorithm, which will be detailed in the following. Then, once a data frame is prepared for transmission by the network layer, it is forwarded to the data link layers, which typically includes the MAC sublayer.

However, if the communication device is based on a data link layer which can induce some delays, between the decision of transmission of the network layer and the effective transmission of the data frame over the air, a non-negligible time might have elapsed. Because of this delay, the data frame when effectively sent over the air may no more be valid. Worse, in some cases, it can create additional messages that can amplify a resurgence of messages leading to a broadcast storm.

Examples of data link layers leading to delays are for example Collision Avoidance based data links, which require an available medium for transmission, or Radio Duty Cycling which allows an intermittent transmission/reception. Other data link layers may cause delay and the invention would be particularly advantageous in case of devices based on such delay prone data link layers.

SUMMARY OF THE INVENTION

It is an object of the invention to propose a method for operating a communication device which alleviates the above problem.

It is another object of the invention to propose a method for operating a communication device and such communication device which prevents no more
5 valid messages to be sent over the network.

It is still another object of the invention to propose a method which reduces the risk of useless broadcast storms caused by not timely sent messages.

The above aims of the invention are solved by a method including the features of claim 1, and a communication device including the features of claim 11.

10 As a consequence, before a data frame can be transmitted, an intervention of the network layer is possible to check whether the data frame can still be sent. If the condition that caused the message to be prepared by the network layer are still met, then the data link layer for example can proceed with the transmission of the message itself. If on the contrary it is detected that the data frame is no more valid,
15 and should not be sent anymore, then the data link layer is able to retrieve such messages from its buffer and cleanse them.

This is particularly beneficial in the context of the Trickle algorithm. Even if a message should have been sent at the time of transmission selected by the network layer, it is preferable to prevent its transmission in case the message was delayed
20 and if the conditions are no more fulfilled. The monitoring can be carried out by the network layer which detects when the conditions are no more fulfilled and in that case, request the data link layer to cancel the transmission of these messages. In addition or in an alternative, the monitoring is systematically carried out by the data link layer, for example for this type of messages, before the effective transmission.
25 Upon transmission by the data link layer, a request for checking the conditions is sent to the network layer.

Thus, at step c), the checking of whether the first condition is still fulfilled may be done upon transmission of the data frame, and/or the checking of whether the first condition is still fulfilled is done upon detection of a second event at the network
30 layer.

To carry out this check, the buffered data frame can be identified by checking at least one of a destination address, a source port and a destination port of the data frame.

Moreover, in an example of the invention, the data frame may be of a first type, and step b) may include deleting any or some previously stored data frames of the first type when buffering the data frame. Thus, it can be configured to destroy any older versions of the data frame or only those carrying similar information.

5 In an embodiment of the invention, the first event can be at least one of the following: reception by the communication device of a data frame carrying information received for the first time at the communication device, for example a message carrying new configuration values, reception by the communication device of a data frame carrying information to be forwarded, for example a message to be
10 relayed to some other devices in the network, arrival of data from a higher layer than the network layer and to be transmitted, for example if the communication device is part of a wireless sensor nodes and a measure from the sensor is received and to be broadcasted in the network.

In another embodiment which can be combined with the previous aspect, the
15 first condition may be at least one of the following: a number of received messages carrying similar information to information carried by the data frame to be transmitted is below a threshold, like for example in the context of the Trickle Algorithm, the validity period of the information carried by the data frame is not expired, if for example a more recent measure has not been received or if each measure is
20 considered valid for a limited duration of time which is not yet expired.

In addition to the previous embodiments, it is proposed in examples that at step c), if the first condition is no more fulfilled, the communication device deletes the data frame from the buffer and prevents its transmission. Thus, the buffer is cleansed from any obsolete data packets before the transmission.

25 In the context of the Trickle algorithm, it is proposed in an embodiment that the method comprises, upon detection of a data frame carrying information received for the first time at the communication device, the network layer prepares a data frame to be transmitted and selects at least partly at random the transmission time of the data frame within a time interval. Then, step b) is carried out at a layer lower than
30 the network layer at said transmission time, in case a number of received messages carrying similar information to information carried by the data frame to be transmitted is below a threshold. Typically, this number can be 1.

Eventually, step c) comprises checking at the network layer that the number of received messages carrying the same information is still fulfilled, and if not,

preventing the transmission of the data frame by deleting said data frame from a lower layer buffer. This can be done at the moment the network layer is about to transmit or at various intervals during the buffering.

As discussed earlier, the communication device link layer may be based on at least of the following: Collision Avoidance system, Discontinuous Transmission, Time Division Multiplexing. These, among other schemes causes delay in the transmission. Thus, the invention would be particularly beneficial for schemes causing delays.

The present invention may also relate to a record carrier carrying a program for carrying out the method of the invention when executed on a computer.

These and other aspects of the invention will be apparent from and will be elucidated with reference to the embodiments described hereinafter.

BRIEF DESCRIPTION OF THE FIGURES

In the following drawings:

Figure 1 is a representation of the OSI model used in the invention,

Figures 2-4, and 6 are time charts for nodes implementing embodiments of the invention,

Figure 5 shows a block diagram representing a network in accordance with an embodiment of the invention.

Figure 7 shows simulation results representing the probability for a node of being updated after a second time interval.

Figures 8A-8B are representative of the update delay results obtained in the network of Figure 5.

Figures 9A-9B are representative of the update delay results obtained in some network implementing the invention.

Figures 10a-f are representative of the number of retransmission required for different values.

DETAILED DESCRIPTION OF THE INVENTION

The present invention relates to a communication device in a network, for example a low-power communication device.

As discussed previously, the invention may be beneficial for a communication device being based on a data link layer which causes delay in transmissions.

A typical example where such delay can be harmful is the Trickle protocol.

Trickle is an efficient algorithm for reducing the amount of communication traffic in low-power wireless networks. Trickle uses a polite gossip approach to disseminate data in an ad-hoc network:

(1) if new information is learned, nodes propagate that information quickly to their neighbors;

(2) if the same information was already sent by the node's neighbors, it will suppress its transmission. This approach enables data to be spread quickly in a network without unnecessary flooding. Trickle is the base algorithm used to form routing tables in the RPL protocol and to disseminate multicast packets in the MPL protocol.

Every node running the Trickle algorithm divides time into intervals of varying length, with a pre-set minimum interval size. Within the second part of each interval, each node uniformly selects a transmission time. Up until that selected transmission time, the node listens to messages from its neighbors. If at least a given, pre-configured, number of packets consistent with its own information have been heard by its neighbors, the node will suppress its transmission within that interval, and at the end of the interval, it will start a new one, for example with twice the size, up until a given maximum. If an inconsistent packet has been received by any node, the following interval will be reduced to the minimum interval size.

The invention focuses on the interaction between the link layer and higher OSI layers. While Trickle is applied at the network layer, low-power networks typically use Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) at the link layer to enable shared access to the wireless media. Furthermore, battery powered devices use radio duty cycling protocols (RDC) to reduce the time the radio is switched on, in to order to maintain longer life time.

CSMA/CA with/without RDC introduces a delay before the transmission of each Trickle packet, and enables re-transmission of packets in case of collisions. If a new packet needs to be propagated, and two nodes try to send the same information at the same time (i.e., their Trickle transmission times are identical), only one will succeed, and the second one will be placed in the link layer outgoing packet queue, for later re-transmission. This re-transmission can have two consequences.

(1) The re-transmission time might fall outside of the Trickle interval boundaries.

(2) Any nodes which were updated by the first packet, will start a new Trickle interval.

5 However, the re-transmission from the second node will come before the third node gets a chance to transmit (it will fall in the first part of the interval, where the node only listens for packets), and may force it to suppress its transmission. Any nodes which are connected only to the third node, will experience larger than expected delays. This effect is strongest for the common, recommended low values
10 of the redundancy constant.

 The aforementioned problem may be solved by introducing Cleansing at the link layer. A Cleansing link layer drops any locally queued Trickle packets if another Trickle packet has been received, or more generally (if k is not 1), if the number of received packets c carrying the same information is above the threshold k . Since a
15 Trickle packet has been received, the queued packet can be seen as obsolete. Cleansing is done based on communication group, as opposed to protocol type and is dependent on protocol parameters, in particular, the received message count c .

 The Cleansing modifications have to be implemented at the link layer. The identification of Trickle packets is done by checking the destination address and the
20 source/destination port.

 Low-power wireless networks, such as networks of ubiquitous sensors, are being built with the aim to be available for extended periods of time, while using as little energy as possible. This includes wireless sensor networks in forests for detecting fires, in pipelines for detecting leaks, on light poles along streets to control
25 luminosity etc. In such resource-constrained devices, wireless transmissions are the largest source of power consumption. Therefore, networking protocols for low-power wireless networks are designed to avoid unnecessary traffic, such as redundant control information or to prevent broadcast storms.

 Trickle has been proposed as an efficient algorithm for controlling traffic flow.
30 It is being used in routing protocols for reducing the amount of control traffic, in multicast protocols for reducing redundant repetitions of data packets and in software update algorithms for managing the propagation of updates. Trickle uses two premises to achieve fast propagation and reduced traffic:

(1) suppressed transmissions when consistent information has been recently propagated by neighboring nodes, and

(2) dynamic transmission rates depending on the consistency of information in the network.

5 The concept of consistency is left to the application layer, which allows the Trickle algorithm to be implemented in different protocols.

10 The Trickle algorithm relies on accurate timing information in order to work as designed. However, various factors can influence this timing and can cause inconsistencies within the protocol. External disturbances can come from the radio medium (packet loss), network (congestion) and locally (data link layer). As a case study, it is considered a MAC layer consisting of unslotted Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) and radio duty cycling. It is shown that due to contended media and CSMA/CA introduced back-offs, nodes can be starved from Trickle updates for long periods of time. This starvation results in large
15 propagation delays and inefficient messaging, making Trickle unsuitable for deadline-critical applications.

20 The Trickle algorithm is used mostly by communication protocols at the network or the application layer. Trickle essentially controls the generation of packets within these protocols. The lower layers are responsible for the actual transmission of the data packets sent by Trickle as can be seen on Figure 1. Fig. 1 shows the Flow of Trickle packets in the Contiki operating system.

25 The data link layer of low-power radios as IEEE 802.15.4, which is the focus in this embodiment, is built of two components - media access control (MAC) and a radio handling protocol. The MAC protocol handles the allocation of the shared medium among nodes and covers retransmissions in case of collisions or packet loss. The radio handling protocol determines the efficient use of the radio during the periods allocated by the MAC protocol.

 A detailed description of the Trickle algorithm and the underlying MAC layer protocols is detailed below.

30 Trickle has two main goals. Firstly, whenever new information becomes available in the network, it must be propagated quickly to all nodes. Secondly, when there is no update, communication overhead has to be kept to a minimum. The Trickle algorithm achieves this by moderating the number of packets that nodes generate with a “polite gossip” policy.

Nodes divide time into intervals of variable length. During each interval a node will send a data packet, only if it has not heard k or more consistent transmissions from other nodes in that interval; otherwise it will stay quiet. Additionally, it will increase the length of its interval, decreasing the rate at which it sends packets.

5 When a node receives inconsistent transmissions, it will reduce the size of its interval, increasing its packet sending rate, in order to quickly resolve the inconsistency.

Trickle only determines when nodes should transmit; the nature of the transmission (broadcast/unicast), the structure of the message, and the exact
10 definition of what is a consistent transmission is given by the upper layers, i.e. the protocols where Trickle is used. For instance, in dissemination protocols, as multicast, transmissions are always broadcasts; a node receives a consistent transmission when a known data packet is received from another node, and an inconsistent transmission is received when a new, unseen data packet is received.

15 The algorithm has four global parameters, which are the same at every node in the network: a threshold value k , called the redundancy constant, minimum (I_{min}) and maximum interval size (I_{max}), and a listen-only parameter (ETA), which defines the size of a listen-only period. By default, $ETA = 1/2$. Furthermore, each node in the network has its own timer and keeps track of three variables: the size of the current
20 interval (I), a counter (c) of the number of consistent data packets received during the current interval, and the transmission time (t) in the current interval.

The behavior of each node is described by the following set of rules. At the start of a new interval a node resets its timer and counter c and sets t to a value in $[ETA \cdot I, I]$ at random. When a node receives a new data packet that is consistent with
25 the information it has, it increments c by 1. When a node's timer reaches time t , it sends a data packet to its MAC layer queue if $c < k$. When a node's interval ends, it sets its interval size to $\min(2I, I_{max})$ and starts a new interval. When a node receives a data packet that is inconsistent with its own information, then if $I > I_{min}$ it sets I to I_{min} and starts a new interval, otherwise it does nothing.

30 Fig. 2 shows the example of three synchronized nodes using the Trickle algorithm ($k = 1, I = I_{max}$).

In the first interval, the transmissions by nodes 1 and 2 are suppressed by the transmission of node 3, while in the second interval, node 2 suppresses nodes 1 and 3.

In Figure 2 an example is depicted of a network consisting of three nodes using the Trickle algorithm with $k = 1$ and $I = I_{\max}$ for all nodes. During the first interval, node 3 is the first node to schedule and broadcast a data packet. The broadcasts of nodes 1 and 2 during that interval are then suppressed. During the second interval the broadcast of node 2 suppresses the other broadcasts.

Note that in the example in Figure 2, the intervals of the three nodes are synchronized. However, in general, the times at which nodes start their intervals need not be synchronized. In practice, networks will generally not be synchronized, since synchronization requires additional communication and consequently imposes energy overhead. Furthermore, as nodes get updated and start new intervals, they automatically lose synchronicity.

The four Trickle parameters can be used to tweak the algorithm behavior according to specific scenarios. For instance, in highly lossy networks, increasing the redundancy constant k would cause nodes to transmit more often, with the intent that the same packet would be propagated to all nodes, in spite of any packet loss.

Similarly, $I_{\min}$ provides a trade-off between speed of propagation and number of packets: lower values of $I_{\min}$ will make nodes transmit sooner, though with an increased risk of collisions, and therefore, additional transmissions. To prevent such scenarios, $I_{\min}$ is usually set to be a multiple of the expected link layer latency in the network. Typical values of the Trickle parameters for various protocols are given in Table 1. In the remainder of this paper, we will focus on broadcast-based data dissemination as the Trickle application protocol, similar to the MPL protocol, with the recommended value for the redundancy constant ($k = 1$).

Table 1. Default values of Trickle parameters in different protocols.

Protocol	k	$I_{\min}$	$I_{\max}$
MPL (control traffic) [9]	10	10 times worst-case link-layer latency	300 s
MPL (data traffic)	1	10 times expected link-layer latency	$I_{\min}$
RPL (DIO) [23]	10	8 ms	8.280 s
CTP [8]	$\infty(0)$	125 ms	500 s

The actual transmission of packets generated by Trickle is left to the MAC layer. Protocols at this layer handle the allocation of the shared media among nodes and cover retransmissions in case of collisions or packet loss. The IEEE 802.15.4 MAC defines two flavors of the CSMA/CA protocol, depending on the operational mode in use: slotted CSMA/CA, used in beacon-enabled modes, where beacons are

sent to synchronize nodes to a super-frame structure; and unslotted CSMA/CA, used in non beacon-enabled modes, where no beacons are sent out and there is no synchronization between nodes. In this embodiment, it is considered an unslotted CSMA/CA, but the same concepts apply to slotted CSMA/CA.

5 In unslotted CSMA/CA, the basic time unit is the back-off period BP, which is related to the transmission time of a frame. Every device maintains two variables for each frame it wants to send: a back-off exponent BE, and a counter for the number of back-offs for the current transmission NB. The ranges of these variables are controlled by three parameters: the minimum back-off exponent BE_{min}, the
10 maximum back-off exponent BE_{max} and the maximum number of back-offs NB_{max}.

Initially, NB = 0 and BE = BE_{min}. Before each transmission, each node first waits for a random number of BPs ranging from 0 to 2BE - 1. After the initial back-off, a node performs a clear-channel assessment (CCA) to determine whether the channel is free. If the channel is free, the node proceeds with the transmission.

15 Otherwise, it increases NB by one, and sets BE to min(BE + 1, BE_{max}). If NB is less or equal to NB_{max}, the entire procedure is repeated. After NB_{max}+1 failed attempts, the frame is dropped from the MAC queue.

The MAC layer of low-power radios often includes a second component next to the CSMA/CA protocol - the radio handling protocol. Radio transceivers are
20 among the biggest sources of energy consumption in low-power wireless devices. Therefore, low-power wireless devices must trade-off between keeping the radio transceiver off, to save energy, and periodically wake up to be able to receive data from their neighbors. During the years, many radio duty cycling (RDC) protocols have been proposed.

25 They can be first categorized into synchronous, where nodes are synchronized with their neighbouring nodes, and asynchronous, where no pre-synchronization is required. Asynchronous RDCs can be further categorized into sender initiated and receiver initiated protocols. Sender initiated RDC protocols give the transmission incentive to the senders: senders wake up receivers to receive a
30 transmission. Receiver initiated protocols give the incentive to the receivers: receivers inform senders when they are prepared to receive a transmission. Finally, hybrid approaches have been developed, which combine features from any of the given categories.

Figure 3 shows, in ContikiMAC, broadcast transmissions that are sent with repeated frames for the full wake-up interval.

As an example, ContikiMAC is considered, where a sender initiated RDC. It is similar to the Coordinated Sampled Listening protocol (CSL), introduced in the IEEE 802.15.4e standard. The basic operational mode of ContikiMAC follows.

By default, every node has its radio turned off. Periodically, at regular intervals of w time units, each node turns its radio on to check for incoming traffic. If a transmission is detected, the radio is kept on until the frame is received. Transmissions are non-periodic, originating from the upper layer(s). Whenever they arrive, a CCA is done to see whether the medium is free. If it is free, the node starts transmitting immediately. Broadcast transmissions should be received by all nodes, irrespective of their wake up intervals. Therefore, a broadcast transmission will always be repeated for w time units (Figure 3), so that each node will at least once turn on its radio during the transmission. This makes broadcasts expensive both in terms of delay and consumed energy.

The main configuration parameter for ContikiMAC is the radio wake-up frequency $1/w$, i.e. how often each node checks the radio for transmissions. This parameter also dictates the maximum duration for each individual transmission w .

Typical configurations include wake-up frequencies of 4Hz, 8Hz and 16Hz, giving wake up intervals of length 250ms, 125ms and 62.5ms, respectively. Reducing the wake-up frequency reduces the energy usage in the network, at the expense of a higher delay.

A common feature of both sender initiated and receiver initiated RDC protocols is that transmissions are not instantaneous, and there is a variable delay between the intent to start a transmission and the actual receipt. In sender initiated RDC protocols as ContikiMAC, the transmission starts almost immediately after it is received from the upper layers, but it is not completed until the receiver performs its periodic wake up to sample the channel. Similarly, in receiver initiated RDC protocols, the transmission is delayed until the sender receives a request from the receiver, which is again periodically scheduled. Finally, in case of collisions, in both cases, CSMA/CA will re-schedule transmissions after a certain back-off period. The delayed completion of a transmission creates a window where upper layer protocols may think that a transmission has been completed, while in fact, it is not. This

causes unintended and inefficient messaging, as the transmission delay and retransmissions may move from one to another Trickle interval.

For example, consider a network consisting of two nodes as represented in Figure 4. They use unslotted CSMA/CA in combination with radio duty cycling at the
5 MAC layer.

Packet transmission is regulated by the Trickle algorithm, with $k = 1$ and $ETA = 1/2$.

Both nodes start a Trickle process at the same time, with consistent information for dissemination. They choose transmission times t_1 and t_2 ,
10 respectively, such that $t_1 < t_2$. Both counters are initially set to zero ($c_1 = c_2 = 0$). At time t_1 , since $c_1 < k$, node 1 sends a packet to its MAC layer. Immediately after, it does a successful clear channel assessment and starts with the transmission of the packet. Node 2 has its next wake-up scheduled at time $t_r > t_2$. Consequently, at time t_2 node 2 has not yet received the broadcast of node 1 and will decide to transmit
15 itself, sending a Trickle packet to its MAC layer. Since at this time the channel is busy, CSMA/CA will delay this transmission until $t_2 + bo$, where bo is the back-off time. At time t_r , node 2 receives the transmission from node 1, setting $c_2 = 1$, making the queued packet in the MAC layer obsolete. However, since there is no link between the MAC queue and the application layer, the pending packet will be sent
20 out at $t_2 + bo$. This effect can be cascaded if multiple nodes exhibit the same behavior. Moreover, it is possible that node 2's broadcast is delayed into its next Trickle interval, as in Figure 4, causing node 1 to suppress its next broadcast due to stale data, disrupting the Trickle process even more.

Fig. 4 shows in detail the MAC layer interference on Trickle timing. Nodes 1
25 and 2 get updated at the same time, and they select transmission times at t_1 and t_2 , respectively. If the reception for node 2 (t_r) is scheduled to be after t_2 , node 2 will queue a Trickle packet at t_2 , even though there is a packet in the air from node 1. Due to CSMA/CA, this packet will be transmitted after the back-off, at time $t_2 + bo$. Note: $k = 1$ in this example.

30 In the example of Contiki operating system for a case study on the impact of MAC interference on Trickle timing. The current implementation of the Contiki operating systems utilizes the ContikiMAC RDC protocol with a radio wake-up interval length of w , together with a slightly modified version of the unslotted CSMA/CA protocol with the following predefined parameter settings. Firstly, the

default parameters $BE_{min} = 0$, $BE_{max} = 3$ and $NB_{max} = 3$, force CSMA/CA to skip the first random back-off. Secondly, the back-off period is equal to the length of the wake-up interval of ContikiMAC ($BP = w$). As w is the worst-case transmission time under ContikiMAC, this makes sure that any retransmissions are attempted after the current transmission has finished. Thirdly, the CCA check is completely delegated to the RDC layer. Finally, the back-off exponent BE is increased only when unicast frames are sent and no acknowledgment is received. Since in data dissemination using Trickle only broadcast packets are sent, for which no acknowledgment is needed, a back-off can only occur due to a failed CCA or a detected collision during a transmission. In both cases, BE remains fixed at one, causing the actual back-off for broadcasts to always remain $BP = w$.

In a potential Scenario, there is a risk of a bottleneck network. Consider now a network of four nodes, with connectivity as in Figure 5. Fig. 5 shows a network consisting of 4 nodes, where node 3 is a bottleneck node.

Again all nodes use CSMA/CA in combination with ContikiMAC and run a Trickle dissemination process. As in the previous scenario, the Trickle process has $k = 1$, $ETA = 1/2$ and $I_{min} = m.w$, where m is at least 2 and is a given constant. Initially, all nodes have consistent information and $I = I_{max}$. Suppose at time 0 nodes 1 and 2 receive an update simultaneously from a closeby source, set $I = I_{min}$ and start a new interval (Figure 6). Fig. 6 shows the suppression of Trickle updates due to MAC layer interference. Nodes 1 and 2 get updated at the same time, and they select transmission times at t_1 and t_2 , respectively, with the periodic channel check for node 2 (t_r) scheduled to be after t_2 . Node 2 will queue a Trickle packet at t_2 . Due to busy media, CSMA/CA will re-schedule the packet until $t_2 + w$. In the mean time, node 3 gets updated and starts a new Trickle interval. The re-scheduled transmission at $t_2 + w$ causes node 3 to suppress its transmission at time t_3 in the first interval. As node 1 and 2 started the second interval earlier than node 3, there is a high probability that they will suppress any future transmissions from node 3.

Node 1 is the first node to schedule a broadcast, which it starts to transmit at time t_1 . As we have seen in the previous scenario, node 2 will schedule a broadcast before receiving node 1's broadcast with probability P_{bo2} . If this happens, the MAC protocol will cause node 2 to delay its transmission until time $t_2 + w$. Before this time, however, node 3 will have been updated by node 1's transmission, and will start a new interval of length I_{min} and schedule a transmission at time t_3 . Now node 2's

transmission follows, suppressing node 3's transmission at time $t_3 > t_2 + w$ and consequently delaying the time that node 4 is updated. In its next interval, node 3 will broadcast only if it starts transmitting before it receives a broadcast by nodes 1 and 2. However, due to the synchronization caused by the Trickle protocol, this has a small probability, as can be seen in Figure 6. In the following intervals the same problem occurs. Only when node 4 eventually transmits its old information, which potentially could take a long time, it will reset node 3's Trickle process and an update will follow.

In general, if node 3 is connected with n synchronized nodes trying to update it, this described scenario occurs with probability P_{bon} . Figure 7 shows a probability and compared it with simulations for different values of m and n in Figure 7. From the plot it is clear that such an event is not rare. Given that such an event occurs, the probability that node 3 will ever broadcast in the following intervals before being suppressed by its neighbors is small, even for $n = 2$.

Fig. 7 is the analytical and simulation results of the probability that node 4 is updated after the second Trickle interval, for different values of m ($I_{min} = m.w$).

Therefore, in such an event, with high probability node 4's update is delayed until it advertises its own old information, resetting the Trickle process of node 3.

This gives an expected delay of approximately $1/2 I_{max} + 3/4 I_{min}$, which is possibly very large since I_{max} is generally large. If node 4 has neighbors suppressing its own transmissions, then the expected delay will be even larger.

In order to reduce the interference of the data link layer on Trickle timing, it is proposed adding a Cleansing mechanism to the MAC layer. If Trickle is treated as a network primitive, known at both the network and data link layer, then some decision making can be done at the data link layer. Assuming that the MAC layer maintains separate queues per destination, whenever a new Trickle packet arrives from the network, the Cleansing MAC will purge any queued outgoing Trickle packets. This will lead to less redundant packets in the network, and will minimize the bottleneck problem from the previous section.

In most cases, purging outgoing Trickle packets improves Trickle performance in terms of messaging and delay, and does not lead to functional incorrectness. It remains consistent with the software design of low-power networks, as any purged packet can be seen as a message loss, and applications are already able to handle

that situation. However, we can identify two scenarios where performance-wise, purging may not be optimal.

The first scenario is when $k > 1$, a purged Trickle message might not be obsolete. However, this should have minimal impact on the network, since only a small fraction of messages within each single-hop broadcast domain will be purged.

The second scenario is when a Trickle message with an old value arrives, and the Cleansing MAC protocol purges an outgoing Trickle message with a new value, increasing the overall propagation delay. This purge can be avoided by making the Trickle consistency check available at the MAC layer, which will cause additional overhead. However, the effect of the purge is minimal, as due to the old message, the Trickle interval of the node with a newer version will be set at I_{min} , which would give a second opportunity for broadcast relatively soon.

In order to confirm the previously analysed negative impact of CSMA/CA back-off on Trickle-based data dissemination, and at the same time evaluate the performance of the Cleansing MAC modifications, Inventors conducted several experiments both in simulation and on a physical test bed. In all cases, we used one application – dissemination of an update using Trickle, implemented in the Contiki operating system. The experiment starts by injecting an update in one (or more) nodes in the network. As the update is propagated, nodes increase their Trickle interval. The experiment ends when all nodes have reached their maximum Trickle interval $I_{max} = 10 \cdot I_{min}$. The delay was measured, i.e. the time required to update all nodes in the network, the number of packets sent during duration of the entire experiment, the number of retransmitted messages by the MAC layer, and the average time messages spent in the MAC layer queue.

It has been simulated the behavior of CSMA/CA with and without cleansing in two scenarios in Cooja, a cross-level simulator for the Contiki operating system. Cooja internally uses the MSPsim device emulator for cycle accurate Tmote Sky emulation, as well as a symbol accurate emulation of the IEEE 802.15.4 CC2420 radio chip. In both cases, all nodes use unslotted CSMA/CA with the default parameters ($BE_{min} = 0$, $BE_{max} = 3$ and $NB_{max} = 3$), and the ContikiMAC radio duty cycling protocol, with a wake-up frequency of 8Hz. As a result, a single-hop broadcast takes $w = 125ms$. The minimum Trickle interval varies between $I_{min} =$

250ms and $I_{min} = 1.75s$, at increments of 250ms, which corresponds to $m = 2, 4, 6, \dots, 14$.

The first scenario follows the bottleneck topology, as shown in Figure 5. An update is inserted at the same time at nodes 1 and 2, and is propagated to the rest of the network using Trickle. Each configuration was simulated 1.000 times. The measured delay for updating node 4 is shown in Figure 8a which shows the Trickle update interval (left y axis) and update delay (right y axis)

Fig. 8 shows the update delay in the bottleneck scenario (Figure 5), using node 1 as a reference ($I_{max} = 256s$, $k = 1$, $ETA = 1/2$). a) shows the Trickle interval in which nodes get updated, with and without Cleansing MAC improvements. b) shows the average delay of the largest 10% of the measurements, and the analytical expected delay. The error bars correspond to the standard deviation.

As expected, without Cleansing, due to the large number of back-offs, the update delay of node 4 is highly variable. The standard deviation, as well as the update delay, peaks at $I_{min} = 0.5s$, and gradually decreases as I_{min} increases. On the other hand, the update delay at $I_{min} = 0.25s$ is fairly constant. The reason for this behaviour is that at $I_{min} = 0.25s = 2 \cdot w$, the contention window for nodes 1 and 2 is equal to the broadcast duration (w). This practically guarantees collisions, and a retransmission from one of the nodes. However, the listen-only period of node 3 will be finished before the retransmission starts, and there is a chance that node 3 will schedule its own transmission before it receives the retransmission. Even if the transmission from node 3 is delayed, it will go through within one or two broadcast periods. However, with $I_{min} = 0.5s = 4 \cdot w$, the contention window for nodes 1 and 2 is still fairly small, giving high probability for back-offs to occur. Then, the retransmission will always fall in the listen-only period of node 3, forcing it to suppress its own transmission. Figure 8b depicts the average measured delay of the worst 10% of the observations. This is a clear indication that harmful back-offs due to CSMA/CA are not uncommon, and that their effects can be detrimental to the Trickle performance.

The update delay then becomes significantly high, in line with the analytical expected delay of $3/4 I_{min} + 1/2 I_{max}$.

On the other hand, the interference is completely resolved when Cleansing is used. In that case, updates are always completed in the second interval, as expected (Figure 8a).

The second scenario consists of 100 nodes, arranged in a 10x10 grid, with 10 meters between two nodes in each axis. A new Trickle event is generated at the top left node. As before, we simulate Trickle with different values for I_{min} . Furthermore, the connectivity range of each node was varied. Each node has a circular coverage area with radius $2+10R$ meters, with R between 1 and 5. Every simulation was repeated 100 times, and all charts show the average values.

Fig. 9 shows the average delay and average number of transmissions in the grid scenario, with and without MAC Cleansing. In this case, using CSMA/CA with Cleansing with $I_{min} = 0.25s$ would require a similar number of transmissions as using regular CSMA/CA with $I_{min} = 1.00s$, while the update delay would be improved by a factor of 2.

Figure 9a shows the update delay, i.e. the time required to update all nodes in the network, using CSMA/CA with and without Cleansing. Since there are no bottlenecks in this scenario, these are comparable. However, the reduction in the number of sent packets is visible in Figure 9b. We can see that the number of transmissions with Cleansing is significantly lower than without Cleansing, while the average update delays are the same.

Figure 10 shows the average number of transmissions and retransmissions as well as the average frame queue time during the entire simulation. As the range of each node grows, fewer messages are required to cover the entire network. Trickle then performs well, suppressing many transmissions (Figure 10a). However, many of the messages are actual retransmissions from the MAC layer (Figure 10b). Since $k = 1$, these are obsolete messages. Furthermore, due to the congested media, frames are left in the queue for a longer time (Figure 10c), often leading to chained attempts for retransmission and further back-offs.

Figures 10d-10f show the impact of using Cleansing. CSMA/CA with Cleansing is aggressive with cleaning the MAC queue, as is visible in Figure 10e. This makes Trickle work as intended even for small values of I_{min} . Additionally, the average queue time is considerably lower compared to the original CSMA/CA.

Fig. 10 shows the average number of transmissions, retransmissions and average frame queue time in the grid scenario, with (d-f) and without (a-c) MAC Cleansing, for different values of I_{min} , $k = 1$ and $ETA = 1/2$.

It is to be noted that the present invention is not limited to the Trickle algorithm and can be applied to other systems including for example Real Time Internet where

a time constraint can be imposed on the data frames transmitted by the communication device.

The invention can be applied for any kind of low-power wireless network which uses the Trickle algorithm. This includes all 6LowPAN based networks, which use RPL for routing and/or MPL for delivering multicast traffic. The invention extends to obsoleteness (delayed forwarding or acknowledgement packets) deriving from the duty cycling and CSMA/CA behavior of the link.

Other variations to the disclosed embodiments can be understood and effected by those skilled in the art in practicing the claimed invention, from a study of the drawings, the disclosure and the appended claims. In the claims, the word “comprising” does not exclude other elements or steps, and the indefinite article “a” or “an” does not exclude a plurality. The mere fact that certain measures are recited in mutually different dependent claims does not indicate that a combination of these measures cannot be used to advantage.

The foregoing description details certain embodiments of the invention. It will be appreciated, however, that no matter how detailed the foregoing appears in text, the invention may be practiced in many ways, and is therefore not limited to the embodiments disclosed. It should be noted that the use of particular terminology when describing certain features or aspects of the invention should not be taken to imply that the terminology is being re-defined herein to be restricted to include any specific characteristics of the features or aspects of the invention with which that terminology is associated.

CLAIMS

1. A method for operating a communication device, the method comprising
 - 5 a) the communication device, upon detection of a first event, preparing at the network layer a data frame to be transmitted at a selected transmission time,
 - b) in case a first condition is fulfilled at said transmission time, the communication device buffering the data frame for transmission at a layer lower than the network layer,
 - 10 c) the communication device checking that the first condition is still fulfilled before transmission of the data frame by the lower layer.
2. The method of claim 1, wherein the first event is at least one of the following: reception by the communication device of a data frame carrying information received
15 for the first time at the communication device, reception by the communication device of a data frame carrying information to be forwarded, arrival of data from a higher layer than the network layer and to be transmitted.
3. The method of claim 1 or 2, wherein the first condition is at least one of the
20 following: a number of received messages carrying similar information to information carried by the data frame to be transmitted is below a threshold, the validity period of the information carried by the data frame is not expired.
4. The method of any of the previous claims, at step c), if the first condition is no
25 more fulfilled, the communication device deletes the data frame from the buffer and prevents its transmission.
5. The method of any of the previous claims, wherein at step c), the checking of whether the first condition is still fulfilled is done upon transmission of the data frame.
30
6. The method of any of the previous claims, wherein at step c), the checking of whether the first condition is still fulfilled is done upon detection of a second event at the network layer.

7. The method of claim 1, wherein step a) comprises, upon detection of a first event, said event being the detection of a data frame carrying first information received for the first time at the communication device, preparing at the network layer a data frame to be transmitted and selecting at random the transmission time of the data frame within a time interval;

wherein step b) is carried out at a layer lower than the network layer at said transmission time, in case a first condition is fulfilled, said first condition being that a number of received messages carrying similar information to information carried by the data frame to be transmitted is below a threshold;

wherein step c) comprises checking at the network layer that the first condition is still fulfilled, and upon detection that the first condition is no more fulfilled, preventing the transmission of the data frame by deleting said data frame from a lower layer buffer.

8. The method of any of the previous claims, wherein the communication device link layer is based on at least of the following: Collision Avoidance system, Discontinuous Transmission, Time Division Multiplexing.

9. The method of any of the preceding claims, wherein step c) includes identifying said data frame by checking at least one of a destination address, a source port and a destination port of the data frame.

10. The method of any of the preceding claims, wherein the data frame is of a first type, and wherein step b) further includes deleting any previously stored data frames of the first type when buffering the data frame.

11. A communication device comprising
a communication stack including

a network layer arranged for detecting of a first event, and for preparing a data frame to be transmitted at a selected transmission time,

a layer lower than the network layer including a buffer, and wherein the network layer is arranged for forwarding said data frame to said buffer in case a first condition is fulfilled at said transmission time,

the communication device being arranged for checking that the first condition is still fulfilled before transmission of the data frame by the lower layer.

1/9

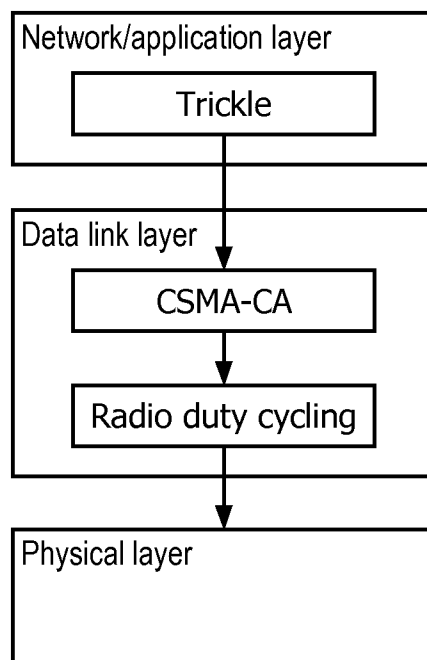


FIG. 1

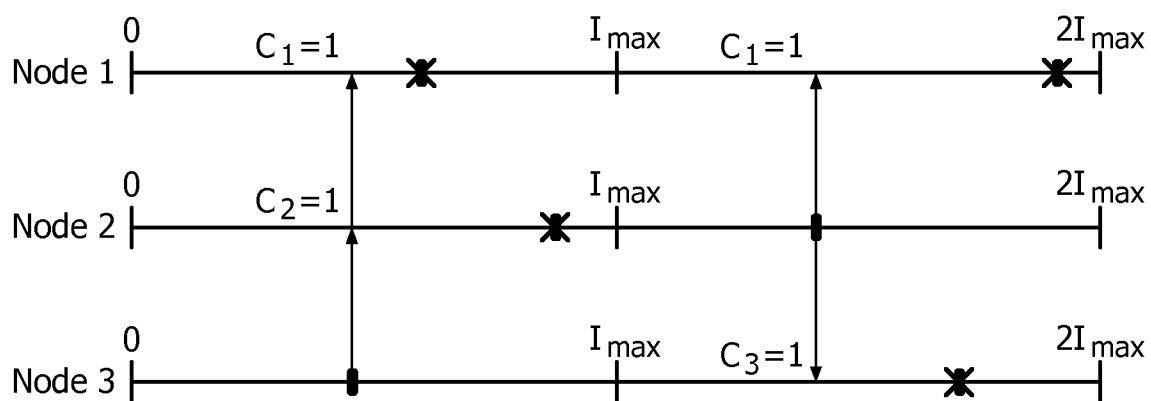


FIG. 2

2/9

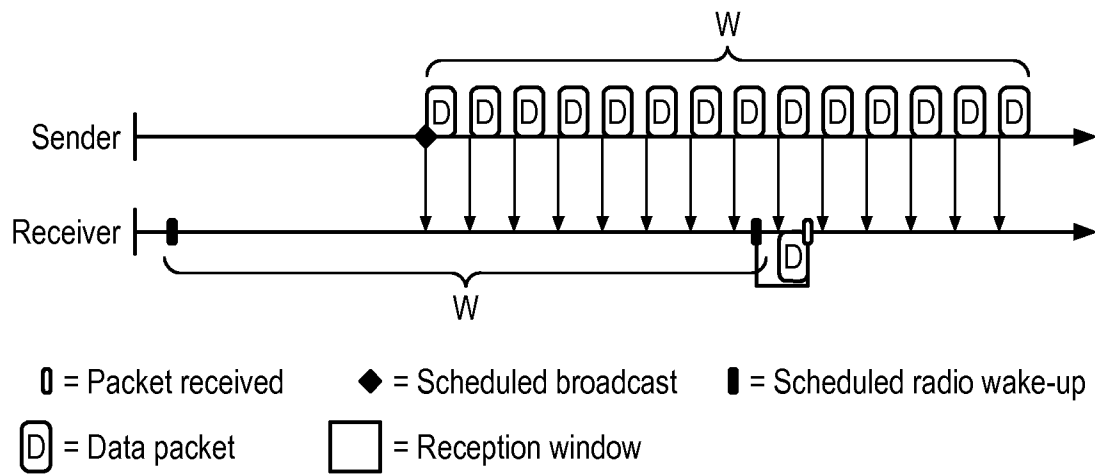


FIG. 3

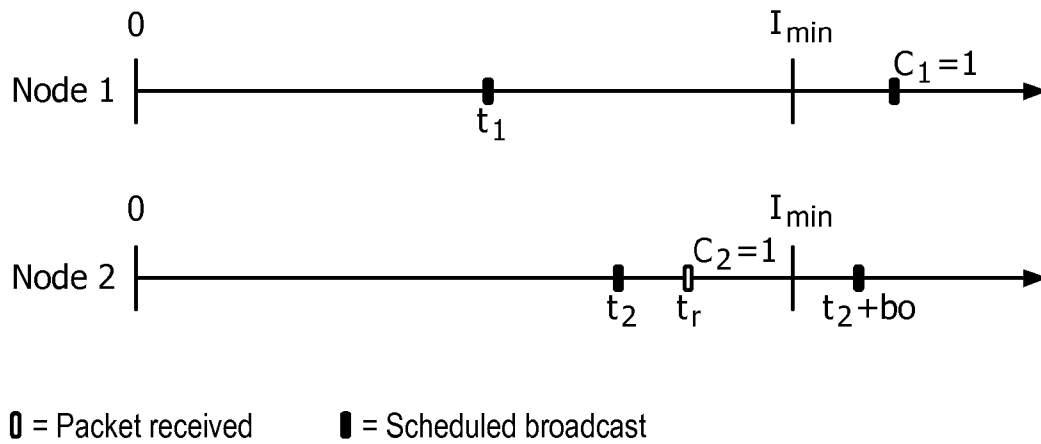


FIG. 4

3/9

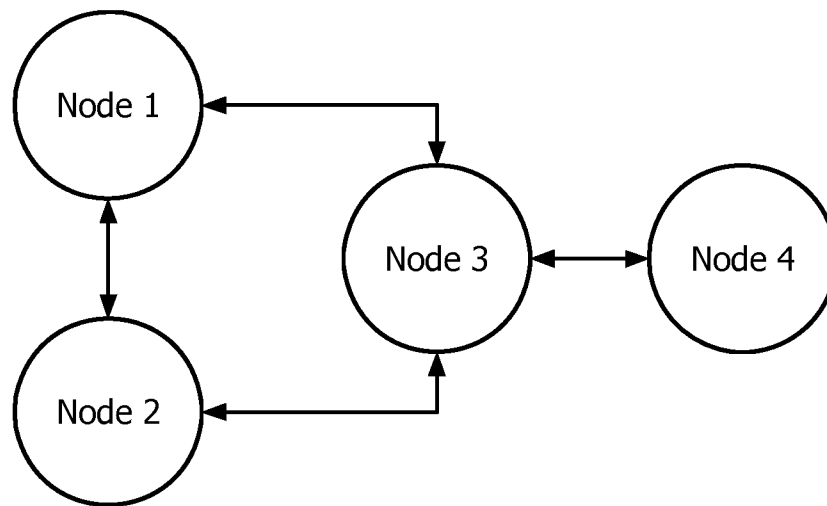


FIG. 5

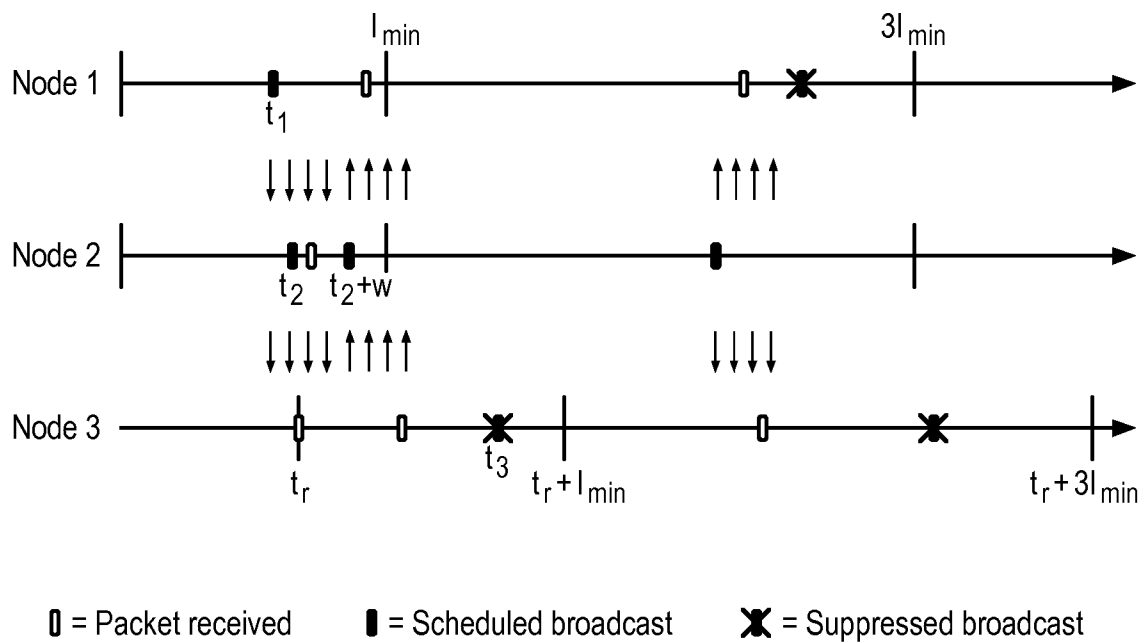


FIG. 6

4/9

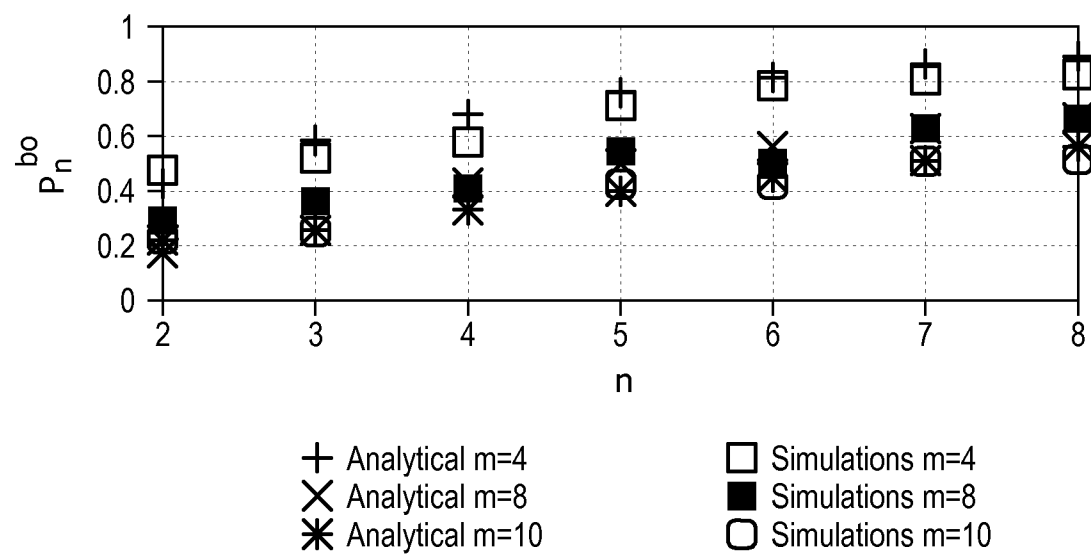


FIG. 7

5/9

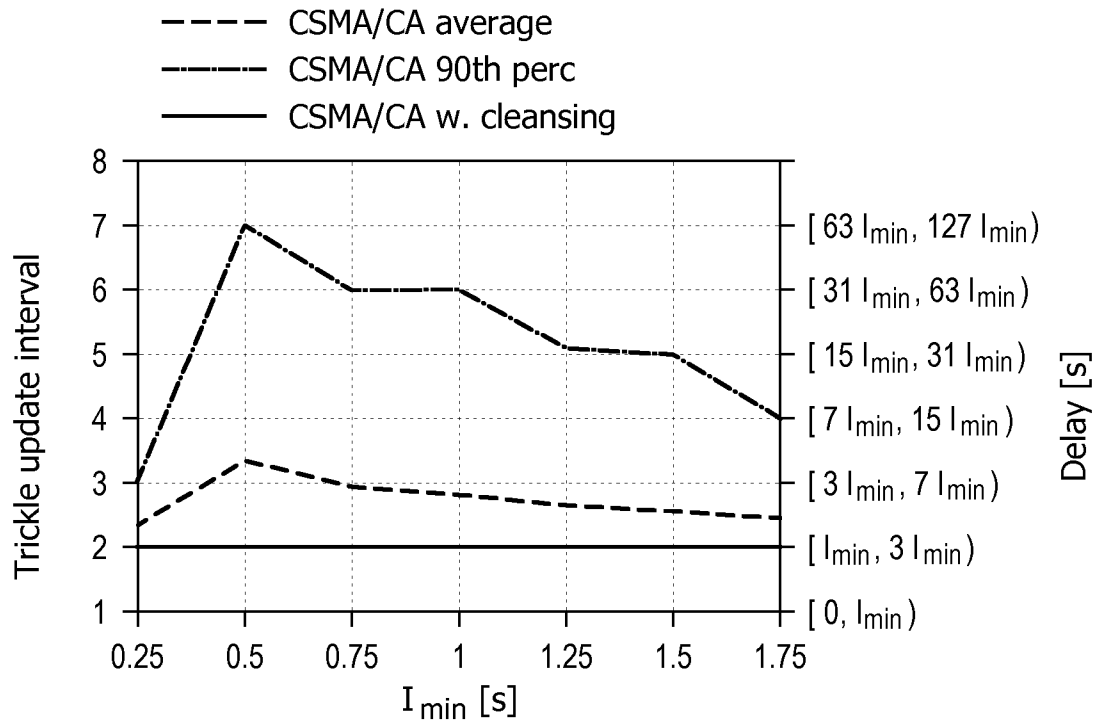


FIG. 8A

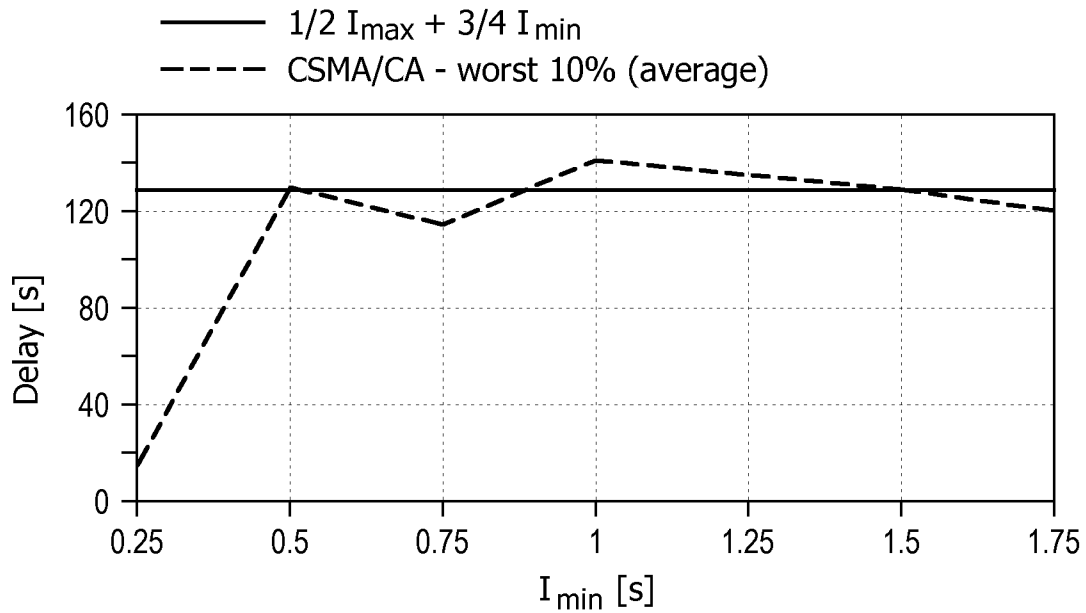


FIG. 8B

6/9

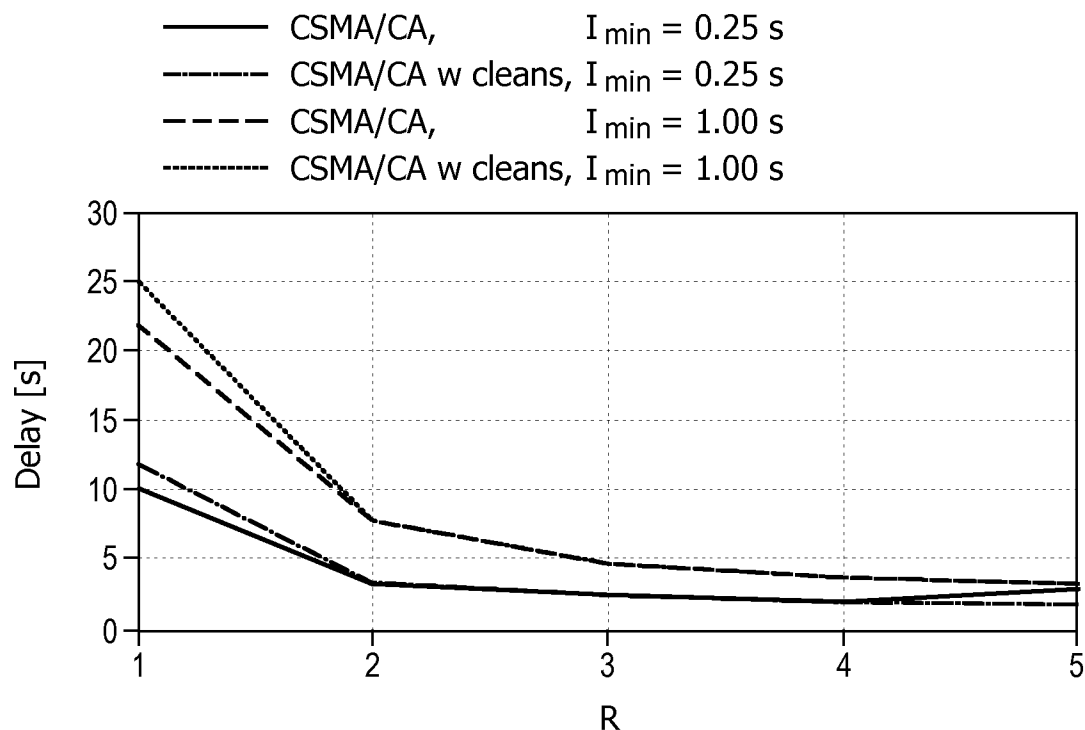


FIG. 9A

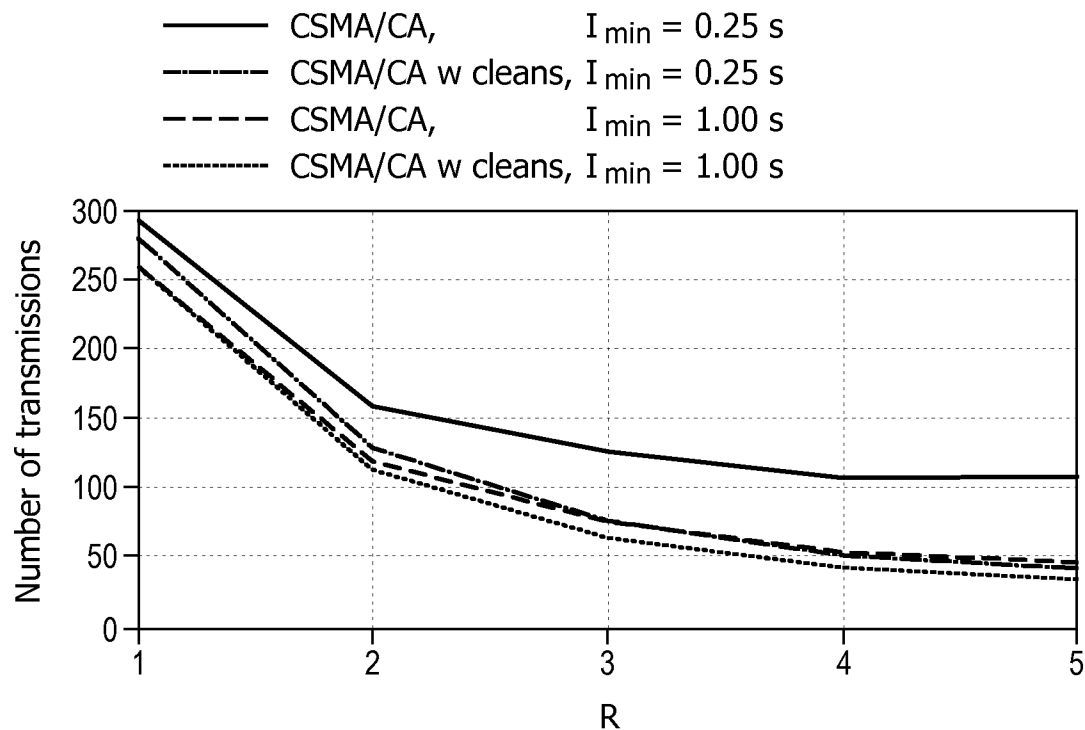


FIG. 9B

7/9

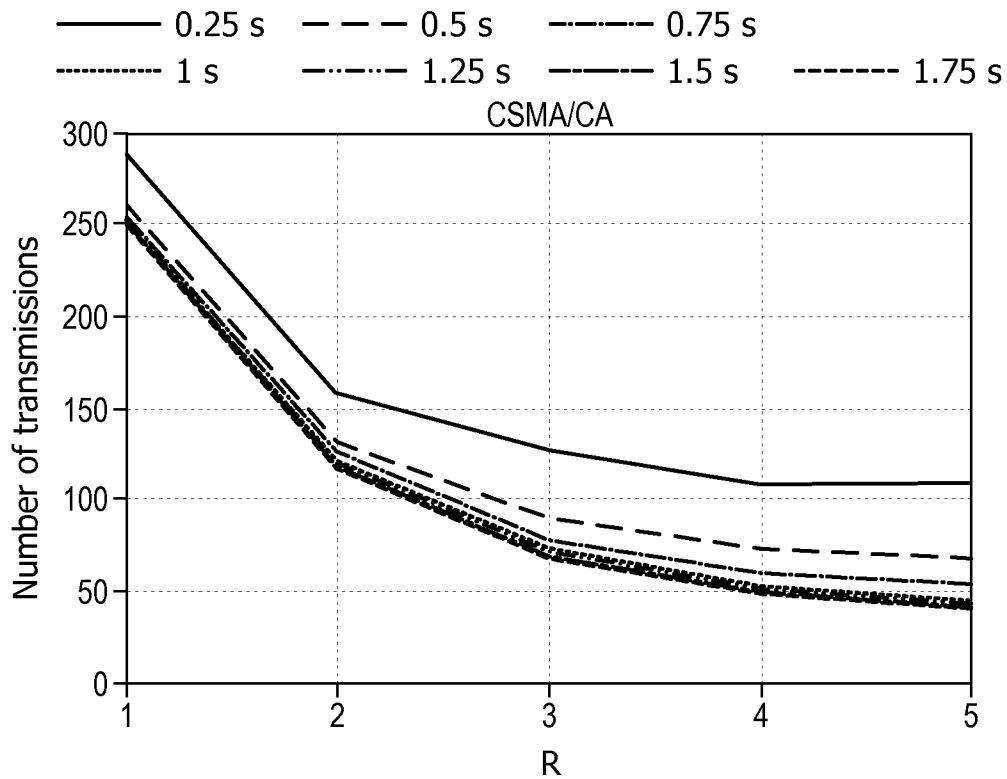


FIG. 10A

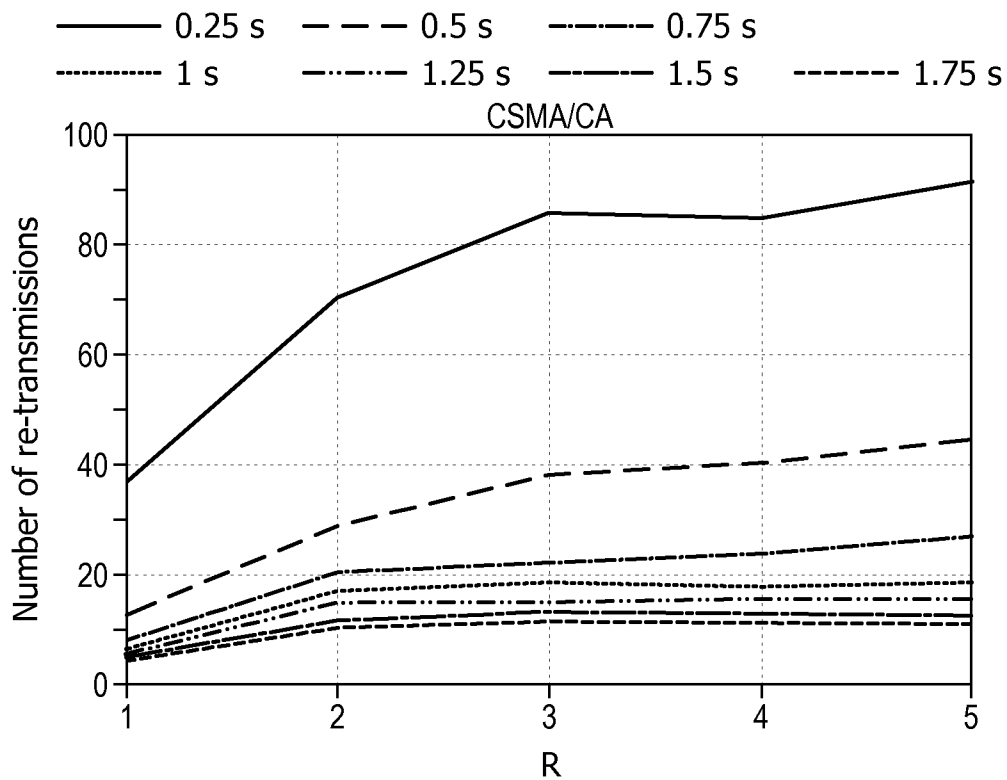


FIG. 10B

8/9

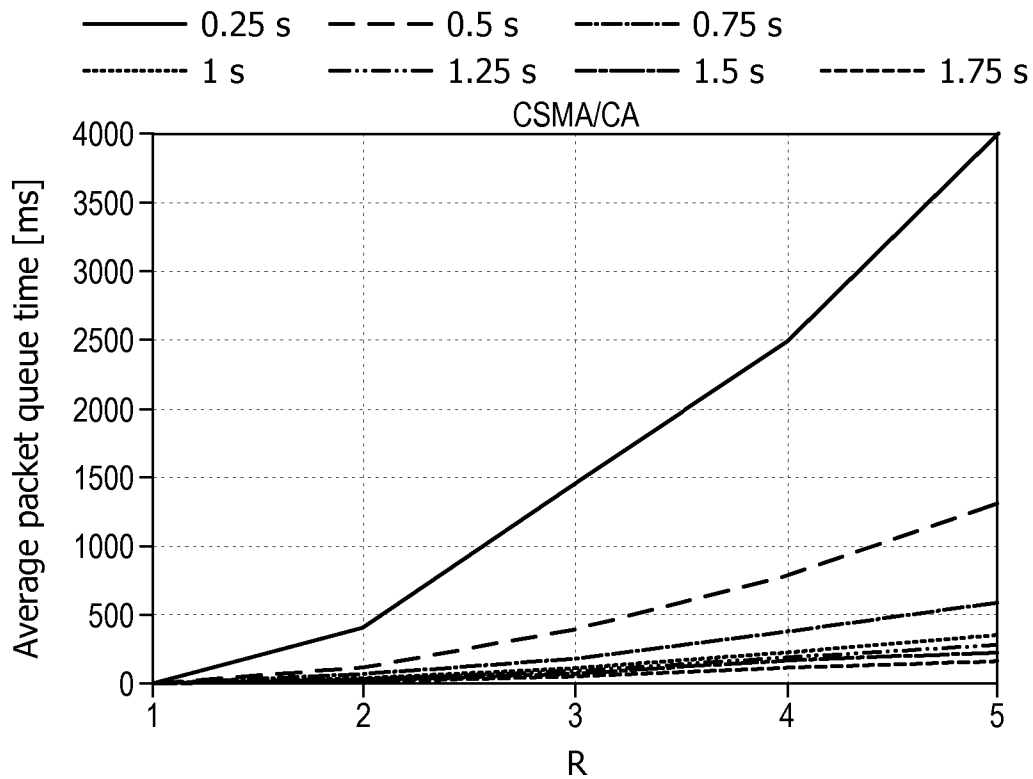


FIG. 10C

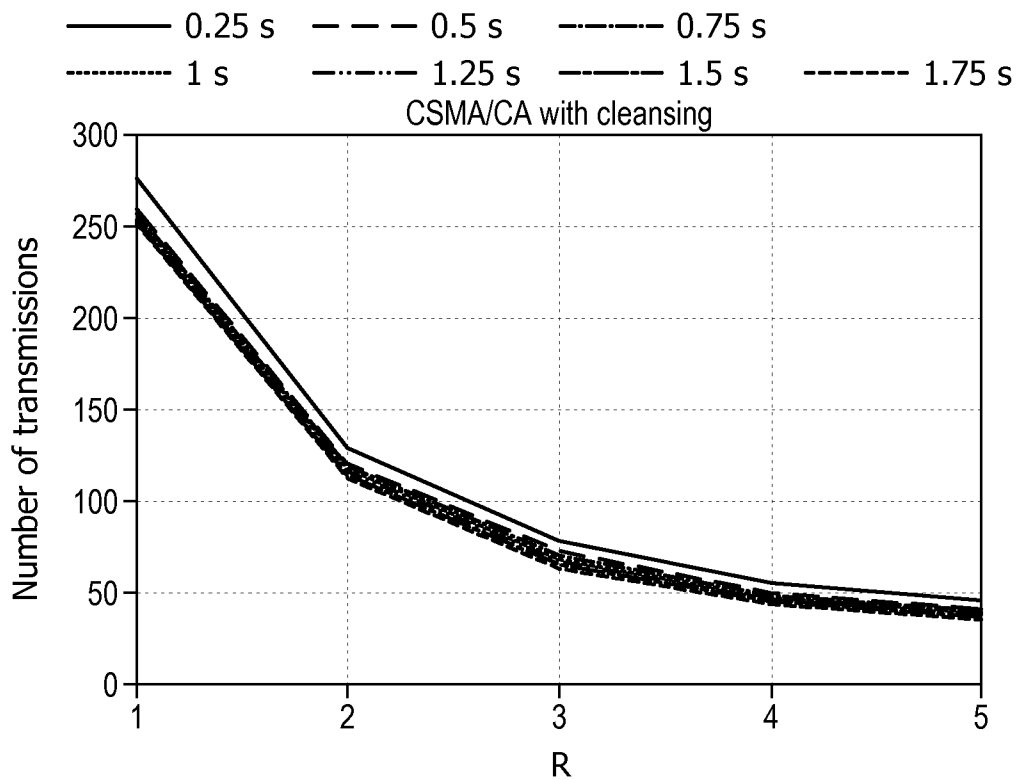


FIG. 10D

9/9

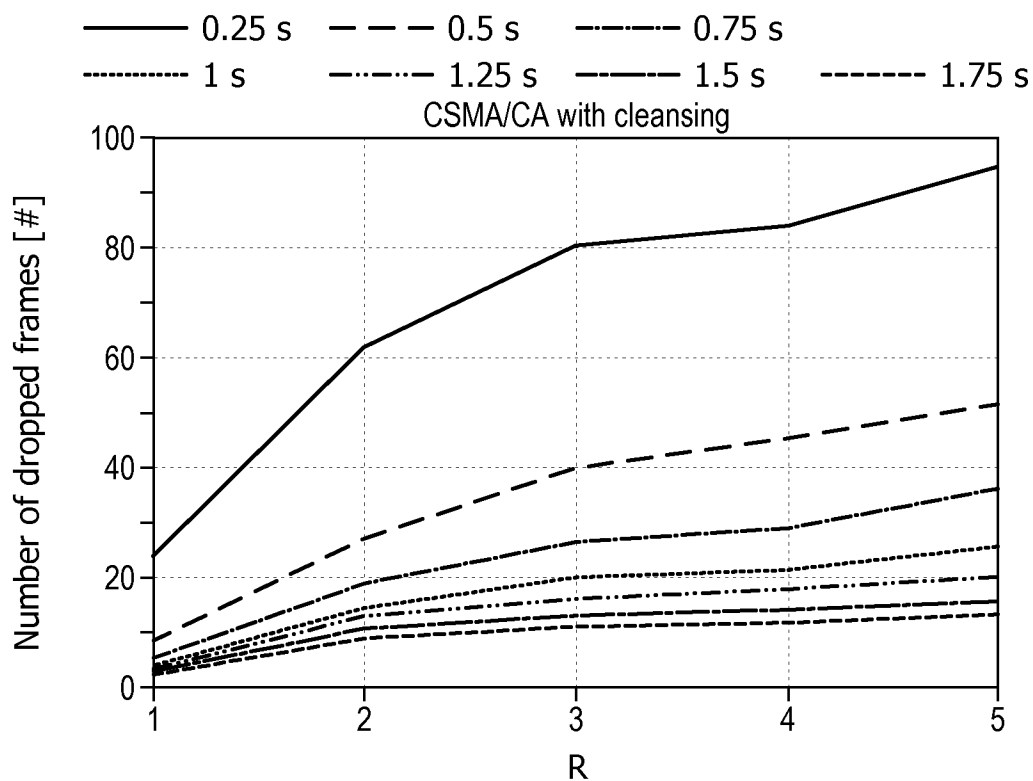


FIG. 10E

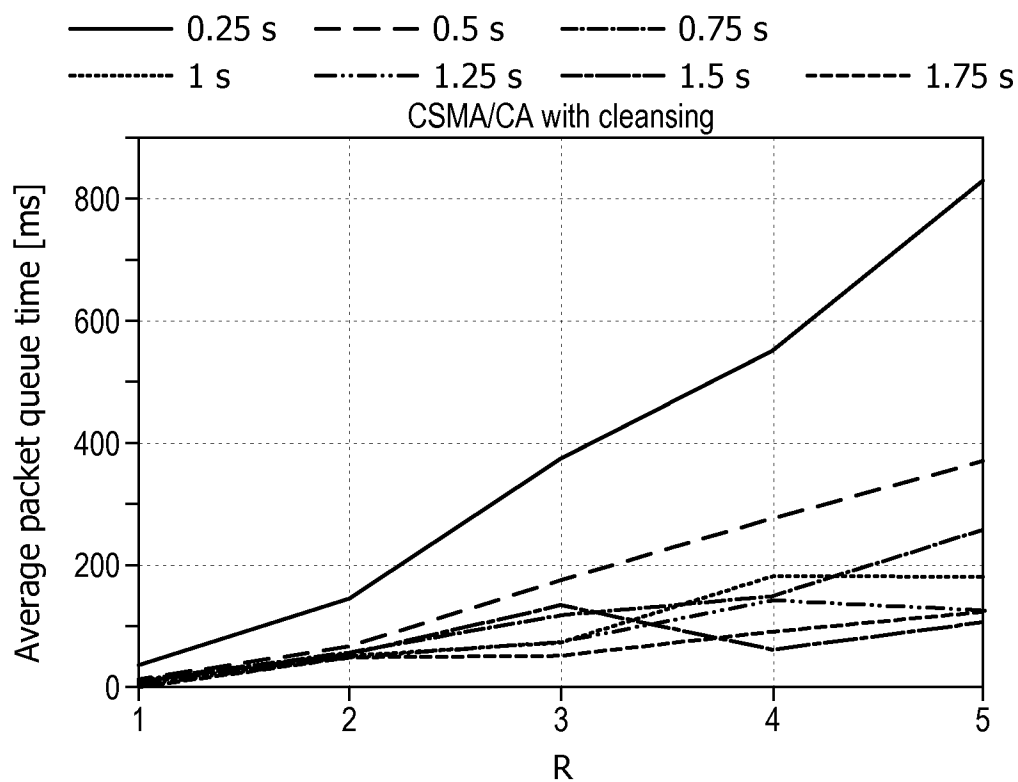


FIG. 10F

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2015/077951A. CLASSIFICATION OF SUBJECT MATTER
INV. H04W28/06
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
H04W

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	PHILIP LEVIS ET AL: "Trickle: A Self-Regulating Algorithm for Code Propagation and Maintenance in Wireless Sensor Networks", USENIX,, 25 March 2004 (2004-03-25), pages 1-15, XP061012375, [retrieved on 2004-03-25] abstract page 1, paragraph 1. Introduction page 4, paragraph 4.Maintenance ----- -/--	1-11



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

1 February 2016

Date of mailing of the international search report

16/02/2016

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Jurca, Dan

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2015/077951

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	HUI CISCO R KELSEY SILICON LABS J: "Multicast Protocol for Low power and Lossy Networks (MPL); draft-ietf-roll-trickle-mcast-03.txt", MULTICAST PROTOCOL FOR LOW POWER AND LOSSY NETWORKS (MPL); DRAFT-IETF-ROLL-TRICKLE-MCAST-03.TXT, INTERNET ENGINEERING TASK FORCE, IETF; STANDARDWORKINGDRAFT, INTERNET SOCIETY (ISOC) 4, RUE DES FALAISES CH- 1205 GENEVA, SWITZERLAND, 24 January 2013 (2013-01-24), pages 1-29, XP015089630, [retrieved on 2013-01-24] paragraph [5.4Trickleparameters] paragraph [7.4Bufferedmessageset] paragraph [10.1;10.2;10.3] -----	1-11
A	DJAMAA BADIS ET AL: "Enhancing delay-based packet forwarding schemes in Wireless Sensor Networks", 2013 4TH ANNUAL INTERNATIONAL CONFERENCE ON ENERGY AWARE COMPUTING SYSTEMS AND APPLICATIONS (ICEAC), IEEE, 16 December 2013 (2013-12-16), pages 12-17, XP032564933, DOI: 10.1109/ICEAC.2013.6737629 [retrieved on 2014-02-11] the whole document -----	1-11