Computer Networks 104 (2016) 79–93

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

TRM-MAC: A TDMA-based reliable multicast MAC protocol for WSNs with flexibility to trade-off between latency and reliability

Ashutosh Bhatia*, R.C. Hansdah

Department of Computer Science and Automation, Indian Institute of Science, Bangalore, India

ARTICLE INFO

Article history:
Received 21 June 2015
Revised 7 April 2016
Accepted 21 April 2016
Available online 10 May 2016

Keywords:
Wireless sensor networks
Media access control
Reliable multicast

ABSTRACT

Multicast in wireless sensor networks (WSNs) is an efficient way to deliver the same data to multiple sensor nodes. Reliable multicast in WSNs is desirable for critical tasks like code updation and query based data collection. The erroneous nature of the wireless medium coupled with limited resources of sensor nodes, makes the design of reliable multicast protocol a challenging task. In this paper, we propose a framework for reliable multicast transmission in WSNs using TDMA-based channel access which works on top of a Multicast Spanning Tree (MST) rooted at the base station. The existing TDMA-based MAC protocols do not provide any mechanism to handle the collision and explosion of feedback messages, and therefore, they cannot be used in the proposed framework to support reliable multicast. To handle this issue, we propose a TDMA-based reliable multicast MAC (TRM-MAC) protocol for WSNs. The TRM-MAC protocol is parametric in the sense that it can be used to trade-off between reliability and delay performance, as per the requirement of the underlying applications. We have analyzed the TRM-MAC protocol to evaluate its delay and reliability performance at different packet loss rates, and have also compared its performance with those of others using simulation study. Both simulation and analytical results show that the TRM-MAC protocol considerably improves the performance of multicast communication in WSNs.

1. Introduction

A WSN consists of a number of sensor nodes, which have limited energy, CPU power, and memory. The sensor nodes may run different applications for different tasks such as event detection, localization, tracking, and monitoring. Such applications should be updated and configured multiple times during the lifetime of the network. An update by transmitting the contents one by one to individual sensor node would be very inefficient and would consume a lot of resources such as bandwidth and energy. In this situation, multicasting can provide an efficient mechanism for updating and configuring the applications running over sensor nodes by reducing the number of transmitted packets. The diffusive nature of the radio medium, also known as the Wireless Multicast Advantage (WMA), makes the multicast operation more effective in WSNs. Another example of multicast communication in WSNs is on-demand data collection, where the base station (sink node) sends a data query to a pre-specified group of sensor nodes asking them to send their sensory data. Moreover, the group of sensor nodes may not be known to the sink node beforehand. An example of such a query is “which sensor nodes have measured the temperature above θ degree?”. A detailed discussion on multicast communication in WSNs can be found in [26].

A WSN is usually multi-hop in nature, and therefore, direct transmission of multicast message from the sink to the sensor nodes is not possible. A straightforward solution to this problem is to let the sensor nodes work as routers. In such a case, every node will relay the received multicast packet to its one-hop neighbors. However, this approach can lead to a large number of redundant transmissions (*network flooding*), and defeat the purpose of using multicast communication. Additionally, forwarding of the same packet in a proximity by multiple nodes, not only causes wastage of energy but also increases the chance of packet collision, and therefore, affects the reliability of multicast transmission. This *network flooding* problem in WSNs is essentially handled by restricting the set of forwarding nodes as much as possible, while still ensuring that all the group members receive the data. To achieve this, a source-rooted multicast spanning tree (MST) is constructed, that contains all destinations (multicast group members) along with a few additional internal nodes (routers). Out of the many possible multicast trees, the tree to be selected, depends upon the optimization goal, viz., minimizing delay, minimizing energy consumption of the nodes or maximizing the bandwidth utilization. The issue of constructing MST with minimal cost is

* Corresponding author.
E-mail addresses: ashutosh.b@csa.iisc.ernet.in (A. Bhatia), hansdah@csa.iisc.ernet.in (R.C. Hansdah).

http://dx.doi.org/10.1016/j.comnet.2016.04.018
1389-1286/

known as the Steiner-tree problem and it is known to be NP-complete [8].

Although having such a multicast tree drastically reduces redundant transmissions, it does not eliminate the possibility of packet collisions among the routers. Moreover, the routers in a proximity would simultaneously start forwarding the received multicast messages and cause a high degree of message collision. This situation of simultaneous data transmission is also known as *correlated-contention* [6].

The above discussion suggests that, in order to achieve reliable multicast in WSNs, the underlying media access control (MAC) protocol, should effectively handle the collisions due to *correlated-contention*. In the literature, time division multiple access (TDMA) and carrier sensing multiple access (CSMA) are the two major approaches for media access in WSNs. Of these two approaches, CSMA does not use any topology or clock information and resolves the contention for every data transmission. Thus, it is highly robust to any change in the network. But, as CSMA is based on the principle of "listen before talk" (LBT), it does not specifically avoid simultaneous transmission or *correlated-contention*. On the other hand, due to its collision-free and energy-efficient properties, TDMA is more suitable as a MAC protocol in WSNs, particularly, in case of *correlated-contention*, where CSMA and its variations are not able to meet the requirements of the applications. There is always a reluctance to use TDMA as a choice for MAC protocol for WSNs due to its synchronization requirement. However, the recent advancement in the design of high-precision and low-cost crystal oscillators, together with large volume of research work to provide efficient protocols for achieving high precision clock synchronization among the nodes in WSNs, has attracted researchers to reconsider TDMA-based solutions for WSNs. Additionally, the clock synchronization is an essential feature of many sensor applications.

Channel errors can occur in a WSN due to background noise and varying properties of communication links. Therefore, merely having an MST to avoid *network flooding* and an efficient MAC protocol to handle collisions due to *correlated-contention*, are not sufficient to ensure reliable multicasting. Any reliable multicast protocol should also provide some mechanism to deal with packet losses. In general, the reliable delivery in multi-hop wireless networks, is handled at either transport layer or MAC layer or at both the layers. One of the first question is, whether to rely entirely on end-to-end acknowledgement at transport layer or to additionally use MAC layer acknowledgements and retransmissions. Due to higher packet loss rate (PLR) in wireless networks, handling packet losses only at transport layer can lead to higher delays and poor bandwidth utilization. Additionally, for high packet loss rate , MAC layer acknowledgements can keep the energy cost within reasonable bounds, whereas the cost for the case with pure end-to-end acknowledgement explodes [14]. Therefore, it becomes necessary to deal with packet loss at MAC layer also in WSNs, and have a balance of end-to-end vs. hop-by-hop reliability.

Most of the wireless MAC protocols use Automatic Repeat reQuest (ARQ) to provide error control for unicast data transmission because of its effectiveness and simplicity. Unfortunately, these ARQ mechanisms cannot be used to ensure multicast transmission reliability, because of following reasons.

- There is no efficient mechanism to resolve the contention between feedback messages sent by the receivers of a multicast packet.
- In case, every receiver sends feedback on receipt of multicast data, the delay incurred to receive the feedback would be very high.

These issues become crucial when the size of multicast group is fairly large. The existing MAC layer protocols for reliable multicasting can be categorized under two major classes based on feedback mechanisms used, viz. positive acknowledgement based (ACK-based) and negative acknowledgement based (NACK-based) approaches. In ACK-based protocols, each time a receiver correctly receives a packet, it returns a positive acknowledgment. Timeouts are used by the sender to detect the loss of packets. In NACK-based protocols, the sender considers packet transmission as successful until it receives a negative acknowledgement (NACK) from a receiver. If the sender receives at least one NACK, it retransmits the packet. The receiver does not send anything on successful reception of the packet. In this approach, the receiver has to check for the lost packets.

In this paper, we propose a TDMA-based Energy Aware MAC (TRM-MAC) protocol for reliable multicasting in WSNs. The protocol is based on a preexisting MST along with a TDMA slot-schedule, specifying the slots at which the nodes can make their transmissions. The TDMA eliminates collisions, overhearing and idle listening, which are the main sources of reliability degradation and energy consumption. To ensure the reliability of a particular transmission at MAC layer, we have used a combination of ACK-based and NACK-based approaches together with TDMA and prioritized-CSMA channel access mechanisms. Furthermore, the proposed protocol is parametric in the sense that it can be used to trade-off reliability with energy and delay as per the requirement of the underlying applications and also to balance the task of reliable multicasting between transport and MAC layers. Note that, in WSNs, all nodes cooperate for a single task, and only one application run at any given time. The TRM-MAC protocol is designed for multicast (sink to sensors) applications, whereas for convergecast applications any MAC protocol available in the literature can be used.

A detailed survey of MAC protocols for WSNs can be found in [12]. Proposed TRM-MAC protocol can be seen as one of the operational mode of a MAC protocol for sensor nodes, that is used only in case of multicast applications. However, the proposed protocol can be integrated with the existing TDMA based MAC protocols, such as IEEE 802.15.4 MAC in beacon mode, to facilitate multicast and convergecast transmissions in WSNs simultaneously.

In summary, the major contribution of this paper is to achieve reliable multicast at link layer which currently is not present in existing TDMA-based MAC protocols. For example, TDMA options present in 802.15.4-2006 MAC and its later revision, do not provide any mechanism to support reliable multicast, i.e., how to handle the acknowledgement message explosion problem and the resulting collision among ACK messages. Additionally, we have proposed a framework for reliable multicast transmission in WSNs based on proposed TRM-MAC which runs on top of an existing MST rooted at the base station.

This paper extends our earlier work proposed in [5], and makes the following core contributions.

- A method to select nodes that use ACK-based approach to improve the reliability performance of the proposed scheme.
- An algorithm to avoid the collision between NACK messages by appropriately deciding the local ordering of nodes that use NACK-based approach. This further improves the reliability of the multicast data dissemination in WSNs.
- Mathematical analysis of the proposed TRM-MAC protocol to evaluate its performance in terms of delay and reliability.
- Simulation of the proposed MAC protocol to evaluate its performance after applying the above mentioned optimization schemes.

The rest of the paper is organized as follows. In Section 3, we discuss the reliability requirement of multicast communication in WSNs. In Section 4, we discuss the proposed framework for reliable multicast in WSNs. The detailed description of TRM-MAC protocol is covered in Section 5. Section 6 analyses the performance of the TRM-MAC protocol in terms of delay and reliability performance. Section 7 discusses the simulation results and Section 8 concludes the paper with suggestions for future work.

2. Reliability requirement of multicast communication in WSNs

In unicast communications, reliability can be defined as the ratio of the number of packets successfully received at the destination to the total number of packets sent by the source. In WSNs, **stochastic delivery** (less than 100 percent reliability) is permitted for unicast communication (sensors → sink), because there is always some redundancy in the data being transmitted to the sink. In case of multicast communication, the transmission of a packet is considered as successful, only when the packet is received by all the multicast group members. According to this definition, the multicast communication (sink → sensors) in WSNs should have **guaranteed delivery** (100 percent reliability), because, in this case, we cannot allow a sensor node to receive only a fraction of the packets transmitted by the sink. For example, when a block of application code needs to be distributed to a set of sensor nodes, losing any packet makes the code block useless. In general, guaranteed delivery is challenging and costly in terms of energy and bandwidth expenditure, especially in multicast communication for WSNs, where a few group members may not receive the transmission either due to the presence of strong localized interference or some obstruction or due to fact that the battery of the nodes might have got exhausted. A relatively relaxed definition of reliable multicast in WSNs can be given as the ratio of the number of sensor nodes that successfully received the multicast data to the total number of sensor nodes in the network belonging to the multicast group. We will use this definition to evaluate the performance of proposed protocol in this paper.

3. Related work

In this section, we discuss some of the existing research work related to reliable multicast in WSNs. The PSFQ (Pump Slowly Fetch Quickly) protocol presented in [33] is designed to deliver a number of segments from a single source node to a subset of receiver nodes in a WSN. The protocol consists of three basic primitives: a pump operation, a fetch operation and a report operation. Using the pump operation, the sink node transmits all the segments one by one, using MAC-layer broadcasts. When a node receives a new segment not yet seen, it stores it in an internal cache. When the new segment is received in-sequence, a node waits for some random time and forwards it. When the segment is received out-of sequence, it is also stored, but instead of forwarding it, the node requests immediate retransmission of the missing segments from any upstream neighbor using a NACK message indicating the missing segment(s) (fetch operation). As soon as the node receives the missing segments, it starts forwarding the segments in-sequence in the pumping mode. Forwarding segments in order ensures that the loss events do not propagate. The report operation is requested by the sink node. The most distant nodes (as indicated by a TTL field in the packet) issue report packets indicating their own address and the received/missing segments. In this way, the sink can judge the progress of the code block dissemination.

The scheme developed in the GARUDA [20] addresses a similar problem as PSFQ, namely, the reliable transfer of block data from a sink to all sensors or a significant part of the network. GARUDA also uses a NACK-based scheme. GARUDA constructs an approximation to the minimum dominating set of the sensor network topology and the members of this set (called core members) act as recovery servers for downstream core members and neighboring non-core members. In RMST (Reliable Multi-Segment Transport)[27], the sink node and all intermediate nodes cache segments and check the cache periodically for missing segments. When a node detects missing segments, it generates a NACK message which travels back to the source along the reinforced path. By investigating different combinations of the above mechanisms, it demonstrated that MAC-layer retransmissions are helpful in case of higher packet loss rates.

The protocols discussed above commonly use NACK-based approach to trigger retransmission and caching of data at multiple nodes to avoid the back pressure all the way up to the sink. The success of these reliable multicast mechanisms highly depends upon the underlaying MAC protocol, which should efficiently handle the collision, packet loss and sleep cycles of the sensor nodes.

In [19], a multicast protocol called BAM (Branch Aggregation Multicast) is presented, which supports single hop link layer multicast and multi-hop multicast via branch aggregation. VLM2 (Very Lightweight Mobile Multicast) [25] is a multicast routing protocol for sensor nodes, which is implemented on top of the MAC protocol. It provides multicast from a base station to any sensor node, unicast connections from a sensor node to the base station. In [9], the authors present an effective all-in-one solution for unicasting, anycasting and multicasting in wireless sensor networks and wireless mesh networks. Furthermore, there are several multicast solutions for WSNs which are based on the geographical position of the sensor nodes in the network [15]. INFUSE [16] is a TDMA-based MAC protocol for data dissemination, and hence it greatly reduces the energy and time taken for flooding. However, the protocol in [16] deals with lossy channels by means of only implicit acknowledgements. TREEFP [24] is another TDMA-based solution which provides energy efficient flooding in WSNs. It provides an algorithm for sleep and wakeup cycles, proposes a method to construct an efficient flooding tree as per the delay requirement of the application, and a heuristic to minimize the TDMA frame size.

Many reliable multicast protocols have been proposed in the literature at MAC layer. The broadcast protocol, Broadcast Medium Window (BMW) [31], uses ACK-based approach. The source sends multicast frame to individual group members on a peer-to-peer basis. The protocol provides high reliability with a very large overhead. If there are n members in the group, the message overhead is n times the size of ACK message plus DATA. In [28], an intermediate approach is employed to reduce the feedback explosion as well as access delay. After sending the multicast data to all receiver nodes using a single transmission, the source sends request to ACK (RAK) message to individual nodes one by one to confirm the status of transmission. This approach imposes less overhead as compared to that in [31] and at the same time it does not require time synchronization between the nodes, which is required in TDMA-based solutions. The protocol proposed in [21] reduces the access delay further by completely eliminating the request to ACK (RAK) messages required to get individual feedback. This protocol uses time division multiple access (TDMA) and requires time synchronization among the receiver nodes to transmit feedback messages at their allotted slots. The broadcast protocols proposed in [21,28,31] could avoid the collision problem and ensure high reliability because of their deterministic nature. But, they still suffer from large access delay and feedback explosion, and are not suitable when the group size is large.

Kuri et al. [18] have proposed a novel mechanism, called leader based protocol (LBP), for reliable multicast with a very less access delay. In LBP, one of the multicast receivers works as a leader which replies with an ACK frame. The LBP provides only probabilistic reliability because reception of data at leader does not always guarantee that the data has been received successfully at all the receivers. The access delay of LBP is the time required to transmit one ACK message, which is independent of the size of the multicast group.

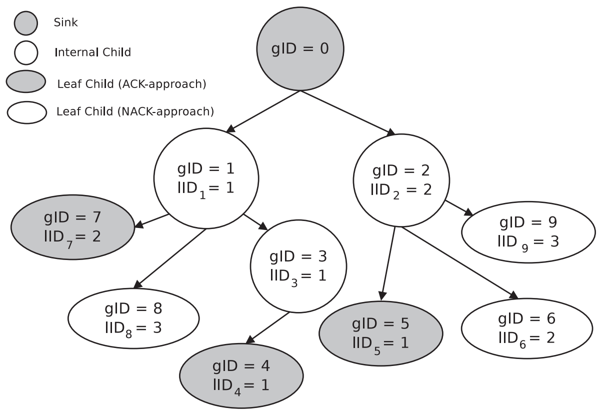

Fig. 1. An example of MST with nodes global ID (*gID*) and local ID (*lID*).

In the design of the above protocols, there is a general perception that access delay needs to be reduced as much as possible while ensuring reliability at the same time. Only in the protocol proposed in [18], reliability has been sacrificed to some extent to achieve the least access delay.

4. Framework for reliable multicast in WSNs

The WSNs have strict resource constraints and are usually deployed by a single organization. Therefore, the traditional layered design approach can be relaxed to provide efficient protocols for services such as reliability. We believe that the problem of reliable multicast in WSNs should be considered as a single problem across all the protocol layers, instead of independently handling it at different layers. Additionally, reliability, delay and energy requirements for a multicast application in WSNs are not static and depend upon current channel conditions, nature of application and availability of resources. Therefore, any solution for reliable multicast in WSNs should support a mechanism to trade-off reliability with delay and energy parameters.

In the following, we propose a framework for reliable multicast transmission in WSNs which consists of following three parts. This framework can trade-off reliability with delay and energy parameters.

1. **Multicast spanning tree (MST) construction**: A spanning tree with base station as the root is constructed to avoid network flooding. The sensor nodes which are not part of the multicast group can sleep while the multicast communication is in progress. The child nodes of an internal node are assumed to be ordered (starting from id 1). Note that the node IDs as per this ordering are local among the child nodes of a particular parent and local ID of a node is not the same as its global ID . We use the notations *gID* as the global ID and lID_i as the local ID of node with i as its global ID, to distinguish between the local and the global identity of the nodes. The terms *internal-child* and *leaf-child* will be used for child nodes which are internal nodes and leaf nodes respectively. The *internal-child* nodes of a parent will always have smaller local ID than the *leaf-child* nodes of the same parent and the local IDs of the child nodes of a parent begin with 1. Fig. 1 shows an example MST, where a node with $gID = 1$ has got three children with $gID = 7, 8$ and 3. Here, node with $gID = 7$ and $gID = 8$ are *leaf-child* nodes, whereas node with $gID = 3$ is an *internal-child* node. Note that the value of lID_3 is less than the value of lID_7 and lID_8.
2. **TDMA slot scheduling:** Each internal node is assigned a data slot in which it can relay the multicast data received from its parent. Only a few leaf nodes (depending upon MAC parameter, *nACK*) are assigned acknowledgement slots (having smaller duration than data slot) in which they can transmit acknowledgement messages to their parent. Note that the internal nodes require multicast scheduling to relay the data, while leaf nodes only require unicast scheduling to transmit the acknowledgements. In the broadcast/multicast scheduling mode, a node can conflict with all the nodes within its two-hop distance. In the unicast mode, a node can conflict with all the nodes which are within one-hop distance of the receiver. A detailed list of conflict relations in wireless networks is given in [22]. We assume that the sensor nodes are synchronized with respect to time. We do not make any assumption about the algorithm used to maintain the synchronization among the sensor nodes. A comprehensive list of synchronization protocols can be found in [29,30].
3. **TRM-MAC protocol:** A MAC frame structure is defined to facilitate the transmission of data and feedback messages (ACK, NACK) by the sensor nodes. To design an efficient link layer feedback and retransmission scheme for reliable multicast, we have essentially used a combination of implicit, positive and negative acknowledgement schemes. The *internal-child* nodes use *implicit acknowledgement* scheme in which they do not explicitly transmit ACK/NACK messages, on receipt of multicast data from their parent. Instead, the acknowledgement is piggybacked at the time when they relay the multicast data received from their parent. This approach ensures that the internal child nodes do not need a separate slot to transmit their ACK, and it also helps to reduce the control overhead. The NACK-based approach is better in terms of access delay as compared to ACK-based approach, because, in NACK-based approach, the nodes only transmit NACK message when they have not received the data; whereas, in ACK-based approach, nodes transmit ACK message when they receive the data, and do not transmit anything when they do not receive the data. But, NACK-based approach requires the receiver to be capable of detecting packet loss. We have used a novel approach to detect packet loss in which a few of the *leaf-child* nodes use ACK-based approach while the remaining leaf nodes use NACK-based approach. The local ID of the nodes using ACK-based approach are always set to be less than that of the nodes using NACK-based approach. As shown in Fig. 1, the local ID of node 7 ($lID_7 = 2$) is less than the local ID of node 8 ($lID_8 = 3$). The transmission of ACK messages by the *leaf-child* nodes using ACK-based approach would trigger the packet loss detection at *leaf-child* nodes using NACK-based approach, which did not receive the data. Note that the *leaf-child* nodes using NACK-approach which have received the data can simply go back to the sleep mode immediately after receiving the data, and do not have to be awake to listen the ACK messages transmitted by other *leaf-child* nodes using ACK-approach. Increasing the number of *leaf-child* nodes, using ACK-based approach, results in larger access delay, but would ensure guaranteed frame loss detection at *leaf-child* nodes using NACK-based approach, and therefore, provides higher reliability. The number of nodes using ACK-based approach *nACK*, can be conveniently varied to tune the reliability of our protocol. At one extreme, all the *leaf-child* nodes can be configured to use ACK-based approach to attain high reliability. To avoid the problem of NACK-explosion, if a *leaf-child* using NACK-based approach receives a NACK message, then it does not transmit NACK message even though it has not received the data.

It is important to note that, in this paper, we have not provided any protocol to construct the MST and to perform the TDMA-scheduling which are required during the part 1 and 2 of the proposed framework respectively. The protocols for the construction of MST in WSNs can be found in [13,23,32,34,35]. Similarly, a com-

Table 1
A list of parameters used related to the TRM-MAC protocol.

Parameter	Description
nACK	Number of *leaf-child* nodes which transmit a positive acknowledgement (ACK) message on successful reception of a multicast data transmitted by their parent.
nRouters	Number of *internal-child* nodes of a parent node.
t_{data}	Time required to transmit a multicast-data at given data rate.
t_{cca}	Time required to perform clear channel assessment.
t_{ack}	Time required to transmit an ACK message at given data rate.
t_{nack}	Time required to transmit a NACK message at given data rate.
gID	global ID of a node in multicast spanning tree (MST).
lID_i	local ID of node with i as its global ID

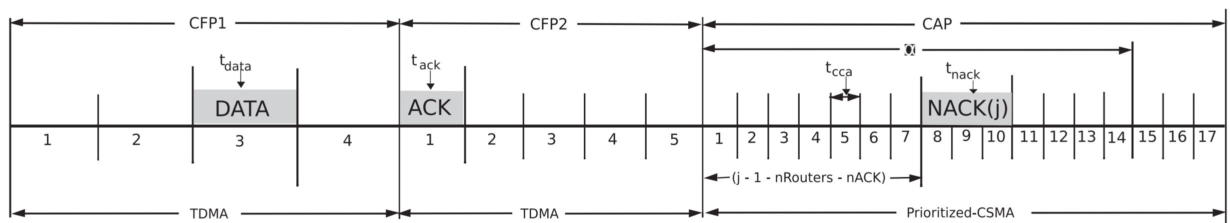

Fig. 2. TRM-MAC-frame structure.

prehensive survey of TDMA slot-scheduling for multihop wireless networks can be found in [17].

5. TRM-MAC protocol

In this section, we describe the proposed TRM-MAC protocol in detail under the following heads. The Table 1 summarizes the description of various parameters related to the proposed protocol, and also, a couple of notations that we have use to describe the protocol.

- Channel access mechanism for the transmission of DATA, ACK and NACK messages.
- Establishment of MAC-frame structure.
- ARQ mechanism to support link layer feedback and retransmission.
- Sleep and wakeup schedule to save the energy of sensor nodes.

5.1. Channel access mechanism

Fig. 2 shows the MAC-frame structure to facilitate the channel access among the sensor nodes to transmit DATA, ACK or NACK messages. As shown in Fig. 2, the MAC-frame is subdivided into three parts, viz. *contention-free-period-1* (CFP1), *contention-free-period-2* (CFP2) and *contention-access-period* (CAP).

CFP1: The channel access mechanism employed in CFP1 portion of the MAC-frame is TDMA, and it is used by the internal nodes to relay the multicast data. The slot-size in this portion is equal to the time required to transmit multicast data (at the current data rate, t_{data}). The number of slots in CFP1 portion depends upon the number of internal nodes in MST and the algorithm used to perform TDMA slot-scheduling for the internal nodes.

CFP2: The channel access mechanism employed in CFP2 portion of the MAC-frame is also TDMA, and is used by the leaf nodes using ACK-based approach to transmit their ACK messages. The slot size in this portion is equal to the time required to transmit an ACK message (t_{ack}), and it is usually smaller than the slot size in CFP1. The number of slots in this portion again depends upon the number of leaf nodes using ACK-based approach in the MST and the algorithm used to perform TDMA slot-scheduling.

CAP: The channel access mechanism employed in CAP portion of the MAC-frame is *prioritized*-CSMA, in which a child node with local id, j using NACK-based approach transmits a NACK message in the CAP portion of MAC-frame, provided it finds the channel idle for $(j - 1 - nRouters - nACK) * t_{cca}$ duration and it has not received the data. The t_{cca} is the time required to perform clear channel assessment (CCA) by any sensor node to ensure that no other node is already transmitting, and *nRouters* is the number of *internal-child* nodes of its parent. The value of t_{cca} is usually smaller than t_{data}, t_{ack} and t_{nack}. The reason for calling this mechanism as *prioritized*-CSMA is that the nodes with smaller local ID value have higher priority to access the channel as compared to the nodes with larger local ID value. The length of this portion would be $(\alpha - 1) * t_{cca} + t_{nack}$, where α is the maximum number of *leaf-child* nodes using NACK-based approach at any internal node, in the network. Note that, the CCA operation may fail in the presence of hidden nodes, and therefore, the proposed *prioritized*-CSMA channel access mechanism does not completely avoid the collision between NACK messages.

5.2. Establishment of MAC-frame structure

To setup the MAC-frame structure among the sensor nodes, each node need to know the time at which the frame starts, and the size and the number of slots in the corresponding portions (CFP1, CFP2, CAP) of the MAC-frame. Note that TDMA-scheduling has already been performed before the start of the multicast transmission by the sink node and the nodes already know their slots in the respective portions of the MAC-frame. The sink node, with slot-id s, can start transmitting multicast data, whenever the application wants to start the communication. The DATA message will also carry the additional information required by the other nodes to calculate the frame boundaries. The sink node will also calculate the frame start time as $t - (s - 1) * t_{data}$, where t is the time of start of the first transmission. Subsequently, on receipt of DATA message from a node with slot id u, the receiver node can calculate the frame start time as $t - u * t_{data}$, where t is the end time of slot u. Note that the current time t is the same at all the sensor nodes, because the clocks are assumed to be synchronized. It is to be noted that the MAC-frame starts at the same time at all the nodes irrespective of the depth of MST. Moreover, in CFP1 portion of the MAC-frame, communication type is broadcast, whereas in the CFP2 portion of the MAC-frame, communication type is unicast. These two parts are scheduled independently for TDMA using a suitable TDMA slot scheduling algorithm such as the one given in [7]. The CAP portion of the MAC-frame does not require any scheduling, and the length of this portion is sufficient to handle the NACK transmission by *leaf-child* nodes, using *prioritized*-CSMA.

5.3. Link layer feedback and retransmission scheme

The feedback approach employed by a node in MST depends upon its local ID. Following three cases are possible with respect to the local ID of a node with $gID = i$.

Case 1 ($lID_i \leq nRouters$)**:** In this case, the node is an *internal-child* and it would use the *implicit acknowledgement* scheme as discussed in Section 4. It could be possible that, an *internal-child* may receive a multicast packet to be relayed from its parent, while previously received packets are still pending to be relayed. In this case, the *internal-child* node can also instruct its parent to delay further transmission for a fixed number of frames (depending upon backlogged packets), to avoid the *buffer overrun* problem. This delaying of transmission by internal nodes would eventually reach to the sink and avoid congestion in the network.

Case 2 ($nRouters < lID_i \leq nRouters + nACK$)**:** In this case, the node is a *leaf-child* and it would explicitly transmit an ACK message (ACK-based approach) in the CFP2 portion of the MAC-frame at its designated slot if it receives the data, otherwise; it does not transmit anything.

Case 3 ($lID_i > nRouters + nACK$)**:** In this case, the node is a *leaf-child* and it will explicitly transmit NACK message (NACK-based approach), if the following conditions are satisfied.

- It did not receive the DATA message in the CFP1 portion of the MAC-frame.
- It received at least one ACK message in the CFP2 portion of the MAC-frame.
- It did not receive any NACK message while waiting to get the access to the channel as per *prioritized*-CSMA, in the CAP portion of the MAC-frame.

Note that the data at a *leaf-child* node can also be received from an internal node which is not its parent, provided the *leaf-child* node is awake. This is due to the fact that the area covered by two different internal nodes would not always be completely disjoint. Similarly, the loss detection at a *leaf-child* could also be due to the reception of an ACK message sent by the *leaf-child* of another parent using ACK-based approach.

Finally, an *internal-child* node would retransmit the data in its designated slot, if it does not receive *nACK* number of ACKs or it receives a NACK message in the CAP portion of MAC-frame against its data transmission in the CFP1 portion of MAC-frame; otherwise, it would assume the transmission as successful and transmit a new data, if available.

5.4. Sleep and wakeup schedule

Traditionally, in multicast transmission, the receiver nodes are assumed to be awake all the time. This is due to the fact that most of the higher layer schemes for reliable multicast do not make any assumption about the underlying MAC protocol. Moreover, the MAC-layer solution for reliable multicast are mainly the extension of stop-and-wait ARQ mechanisms for unicast transmission on top of CSMA based MAC protocol. This does not allow the nodes to sleep when they are not expecting any data and save power.

In the proposed MAC-frame structure, an internal node needs to be awake only during the following time-slots within the MAC-frame.

- At its own data transmission slot to relay the multicast data and the data transmission slot of its parent node and its *internal-child* nodes, to receive the multicast data, in the CFP1 portion of the MAC-frame.
- During the ACK transmission slots of its *leaf-child* nodes using ACK-based approach in the CFP2 portion of the MAC-frame.
- From the beginning of the CAP portion of the MAC-frame till it receives any NACK message, only if it has received *nACK* number of ACKs in CFP2 portion. Otherwise, it can completely sleep during the CAP portion.

A leaf node using ACK-based approach needs to be awake only during the following time-slots within the MAC-frame.

- At data transmission slot of its parent node to receive the multicast data, in the CFP1 portion of the MAC-frame.
- During its ACK transmission slot in the CFP2 portion of the MAC-frame, provided it has received the data.

Finally, a leaf node using NACK-based approach needs to be awake only during the following time-slots within the MAC-frame.

- At data transmission slot of its parent node to receive the multicast data in the CFP1 portion of the MAC-frame.
- During the ACK transmission slots of other *leaf-child* nodes of its parent using ACK-based approach, in the CFP2 portion of the MAC-frame, provided it has not received the data.
- During the CAP portion of the MAC-frame, provided it has not received the data and received at least one ACK message in the CFP2 portion of the MAC frame.

5.5. Selection of nodes using ACK-approach

Although, we do not put any restriction on the selection of nodes using ACK-approach and NACK-approach among the *leaf-child* nodes of an internal node, the reliability performance of the protocol can vary across various possible selections. If there exists a *leaf-child* node with NACK-approach which is not a one-hop neighbor of any other *leaf-child* node using ACK-approach, then it would not be able receive the ACK transmissions, and would fail to detect the packet loss. The problem of selecting nACK number of nodes which would use ACK-approach out of all *leaf-child* nodes of a node j can be viewed as a slightly modified version of the classical maximum coverage problem (MCP) in computer science [11].

Let $S_i^{'}$ be the set of one-hop neighbors of a *leaf-child* node i ($gID = i$) of a node j, such that these neighbors are also the child nodes of node j and $S_i = S_i^{'} \bigcup \{i\}$. Then we define the modified maximum coverage problem as follows.

> **Modified Maximum Coverage Problem (MMCP):** *Given node j, a number $nACK \leq m$ and a collection of sets $S = \{S_1, S_2, \ldots, S_m\}$, where m is the number of child nodes of node j and S_i is defined as above, we have to select a subset $S^{'} \subseteq S$, such that $\left|S^{'}\right| = nACK$ and the number of elements covered by $S^{'}$, i.e., $\left|\bigcup_{S_i \in S^{'}} S_i\right|$ is maximized.*

The maximum coverage problem is known to be NP-hard. But, in MMCP the parent node need to check only ${}^{m}C_{nACK}$ combinations, which is a $nACK^{th}$ degree polynomial in m, and therefore, the computational complexity of MMCP is of the order of $O(m^{nACK})$. Although the computational complexity of MMCP is polynomial, it would not be suitable for sensor nodes with limited computational resources to find $S^{'}$, since the value of *nACK* required to cover up the entire set of child nodes of j may be quite large, and may also approach $m/2$. Note that, the value of ${}^{m}C_{nACK}$ would be maximum when $nACK = m/2$.

Here, we use a greedy algorithm to select the *nACK* number of *leaf-child* nodes of a node *j* which would use the ACK-based

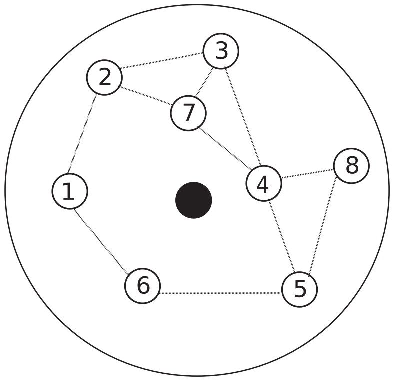

Fig. 3. An instance of modified maximum cover problem with $nACK = 2$.

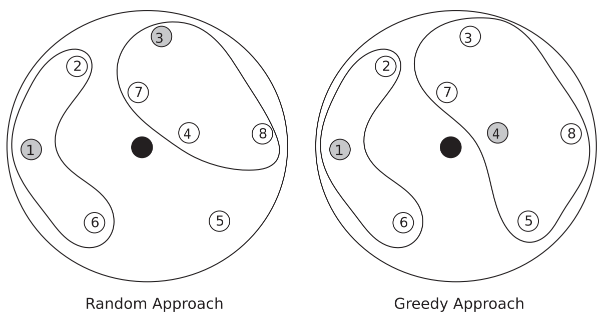

Fig. 4. Comparison between random and greedy approaches to select the nodes which would use *ACK-approach* ($nACK = 2$).

approach. The proposed algorithm chooses a *leaf-child* node i of a parent node j in each iteration, such that the set S_i contains the maximum uncovered *leaf-child* nodes of node j and repeat the same procedure for *nACK* number of times. An uncovered *leaf-child* node, at the beginning of a particular iteration is defined as the node which are not part of the set $\bigcup_{k\in C} S_k$, where C is the set of *leaf-child* nodes that have already been selected as the nodes which would use ACK-approach. The complexity of the proposed algorithm is $O(m^2 \log m)$), as there can be at most m iteration and each iteration may require at most $m \log m$ steps to sort the nodes in order to select the node i such that S_i contains the maximum uncovered *leaf-child* nodes. Note that, the number of uncovered nodes in S_i may change after each iteration and therefore the sorting has to be performed in every iteration.

Consider an internal node whose *leaf-child* nodes are scattered in every direction, as shown in Fig. 3. In this case, $S_1 = \{1, 2, 6\}$, $S_2 = \{1, 2, 3, 7\}$, $S_3 = \{2, 3, 4, 7\}$, $S_4 = \{3, 4, 5, 7, 8\}$, $S_5 = \{4, 5, 6, 8\}$, $S_6 = \{1, 5, 6\}$, $S_7 = \{2, 3, 4, 7\}$, $S_8 = \{4, 5, 8\}$. Note that the ID given along with the nodes in the Fig. 3 are global (gID), as local IDs of the nodes can be decided only after the selection of nodes using ACK-approach and NACK-approach.

Fig. 4 shows the results of both the approaches (random and greedy) for the problem instance shown in Fig. 3. By random approach, we mean that all the combinations of *nACK* number of nodes out of m nodes have the equal probability to be selected. We can see that the proposed greedy approach is able to cover all the leaf-nodes of the parent whereas in case of a random selection, node 5 will not be able to receive the ACK messages sent by node 1 or node 3.

5.6. Deciding local ordering of leaf-child nodes of an internal node

As we mentioned in Section 4 that the child nodes of an internal node are locally ordered (starting from id 1) by its parent. According to this local ordering, the *internal-child* nodes of a parent would have smaller lID value than the *leaf-child* nodes, and similarly, the *leaf-child* node using *ACK-approach* would have smaller lID value than the *leaf-child* nodes using *NACK-approach* (see Fig. 5).

As explained earlier, the channel access mechanism used by the *leaf-child* nodes using NACK-approach in the CAP portion of the MAC frame is prioritized-CSMA. In prioritized-CSMA access, the time when a node can get the access to the channel depends upon its relative position in the local ordering, defined by its parent. The nodes with smaller *lID* value get higher priority to access the channel as compared to the nodes with larger *lID* value. As we mentioned in the Section 5, in the presence of hidden nodes, the CCA operation may fail, and consequently, the transmission of NACK messages by the leaf nodes of a parent may collide.

Fig. 6 depicts the situation for $nACK = 2$ and all the leaf-nodes receive multicast data sent by the parent except nodes 3 and 5. The ID given to a node in Fig. 6 refers to its local ID. Fig. 7 shows the sequence of actions taken by the receiver nodes for the example given in Fig. 6. Note that both node 1 and 2 will send ACK messages since $nACK = 2$. After listening an ACK message from node 2, node 3 will transmit an NACK message at the beginning of the CAP portion. Since node 3 and 5 are not in each others range, node 5 would find the channel idle during slots 1 to 2, while node 3 is actually transmitting NACK message. Therefore, node 5 will start transmitting NACK message at slot 3 which would collide with NACK transmission of node 3.

The problem of providing local IDs to the nodes using NACK-approach can be viewed as the problem of assigning slots to the nodes within CAP portion, in which they can start their NACK transmission. These slots should be assigned in such a way that the nodes which are not one-hop neighbors of each other should not get the slots within each others transmission duration.

Formally, we define the above problem as follows.

Let V be the set of nodes using NACK-approach and N_v be the set of one one-hop neighbors in V for each node $v \in V$. We use the global IDs of the nodes to describe this problem. The problem is to find a mapping $\mathcal{F} : V \to \{1..|V|\}$ such that $\sum_{u\in V}\sum_{v\in V} \mathcal{G}(u, v)$ is minimum, where $\mathcal{G} : V \times V \to \{0, 1\}$ is defined as,

$$\mathcal{G}(u, v) = \begin{cases} 1, & u \notin N_v \text{ and } |\mathcal{F}(u) - \mathcal{F}(v)| <= s_{nack} \\ 0, & \text{otherwise,} \end{cases}$$

where s_{nack} is the number of CAP slots in t_{nack} duration.

We propose an algorithm to be used by a parent node to decide the local IDs of its child node which use NACK-approach. The local ID of nodes using ACK-approach will always be between $nRouters + 1$ to $nRouters + nACK$. Therefore, a value of $nRouters + nACK$ need to be added to the local ID of the nodes assigned by the algorithm, to get the actual local ID of the nodes using NACK-approach.

The proposed algorithm uses a greedy based approach to minimize the function $\sum_{u\in V}\sum_{v\in V} \mathcal{G}(u, v)$, where V is the set of nodes using NACK-approach. The algorithm selects nodes one by one and assigns them local ID in the order they have been selected. Let Q_i be the set of nodes that have been selected at the end of i^{th} step. Then, a node w from the set $V - Q_i$ would be selected in the $i+1^{st}$ step such that $\sum_{u\in Q_i}\sum_{v\in Q_i} \mathcal{G}(u, v) + \sum_{u\in Q_i} \mathcal{G}(u, w)$, is minimum, i.e., the addition of node w would cause minimum increment to the value of $\sum_{u\in Q_i}\sum_{v\in Q_i} \mathcal{G}(u, v)$. In other words, the selected node will have maximum number of neighbors in the set $Q\,|_{i-t_{nack}}^{i}$, where $Q\,|_{i-t_{nack}}^{i}$ is the set of nodes that have been assigned local ID between $i - t_{nack}$ to i.

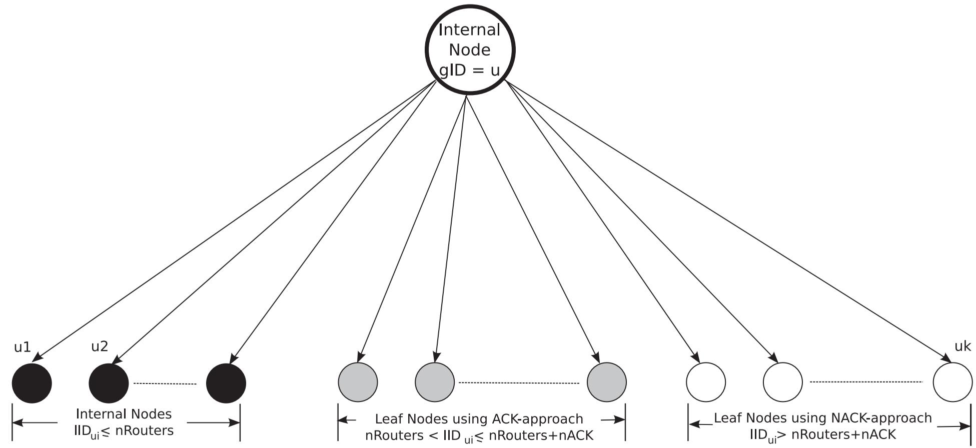

Fig. 5. Local ordering of child nodes of an internal node with $gID = u$ that has k children with $gID = u_1, u_2 ... u_k$.

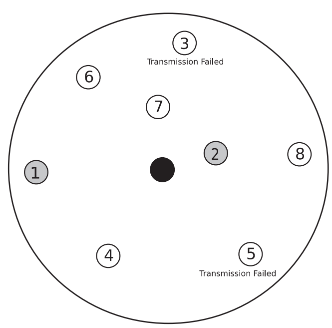

Fig. 6. An example scenario.

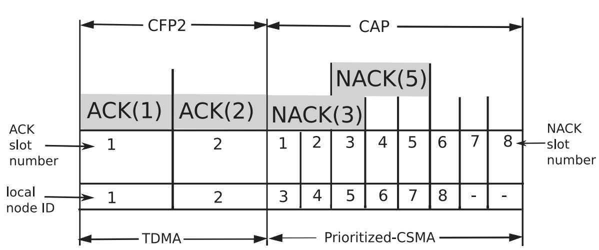

Fig. 7. Collision of NACK messages transmitted by node 3 and node 5.

It could be possible that there exist two or more nodes (say w' and w'') in $V - Q_i$ such that $\sum_{u \in Q_i} \mathcal{G}(u, w') = \sum_{v \in Q_i} \mathcal{G}(u, w'')$. In such a case, we select the node w that has got maximum number of neighbors in the set $V - Q_i - w$, i.e., $|\{v : v \in V - Q_i - w \text{ and } v \in N_w\}|$ is maximum. The selection of a node in this manner at step $i+1$ ensures that the nodes which are left after the selection, will have higher probability of being neighbor of newly added node. This in turn helps to reduce the amount of increment to the function $\sum_{u \in Q_{i+1}} \sum_{v \in Q_{i+1}}$ at the step $i+2$. The proposed algorithm for the assignment of local ID to the nodes using NACK-approach is given in Algorithm 1.

Table 3 shows the step by step execution of Algorithm 1 for the example given in Fig. 6. The set of one-hop neighbors in V for each node $v \in V$ in example 6, is given in Table 2

We can see that the value of $\sum_{u \in V} \sum_{v \in V} \mathcal{G}(u, v)$ for the local ordering (7,3,6,8,5,4) generated by our algorithm is 5, whereas, for the initial local ordering (3,4,5,6,7,8), the value is 7. Note that the optimal value of $\sum_{u \in V} \sum_{v \in V} \mathcal{G}(u, v)$ is restricted by the degree of nodes in the input graph. For example, in this case, the degree of node 4 is only one, and therefore, we cannot avoid the conflict of node 4 with at least two other nodes. Similarly, the degree of node 5, 8 and 6 is 2, and therefore, the optimal value for the above example is also 5 which is the same as the value achieved by our algorithm.

Algorithm 1: Algorithm to calculate local ID of the child nodes

procedure Local-Ordering(G, s_{nack})
$i = 1$ // index for local ID
create a list Q // Initially list Q is empty
while $V - Q$ is not empty **do**
create two lists $L1$ and $L2$
for Each node w in $V - Q$ **do**
if $i \leq s_{nack}$ **then**
if $|N_w \cap Q| \geq |N_v \cap Q|, \forall v \in V - Q$ **then**
select w and put it in the list $L1$
end if
else
if $|N_w \cap Q^i_{i-s_{nack}}| \geq |N_v \cap Q^i_{i-s_{nack}}|, \forall v \in V - Q$ **then**
select w and put it in the list $L1$
end if
end if
end for
for Each node w in L1 **do**
if $|N_w \cap (V - Q)| \geq |N_v \cap (V - Q)|, \forall v \in L1$ **then**
select w and put it in the list $L2$
end if
end for
select a node w from $L2$ such that
$NodeID(w) < NodeID(v), \forall v \in L2$
localID$(w) = i$
Add w to Q and remove it from V
i = i +1
Empty L1 and L2
end while

6. Performance analysis of TRM-MAC protocol

In this section, we analyze the performance of TRM-MAC protocol in terms of latency and reliability parameters. Throughout the analysis, we have used a number of different parameters and these are together collected in Table 2 for easy reference.

6.1. Latency analysis

Let $\mathcal{T}$ be the time when a multicast transmission is received by all the nodes in MST. Our aim is to find the expected value of

Table 2
Neighbor table for nodes in $\in V$

Node Id u	N_u
1	{4, 6, 7}
2	{3, 5, 7, 8}
3	{2, 6, 7}
4	{1}
5	{2, 8}
6	{1, 3, 7}
7	{1, 2, 3, 6}
8	{2, 5}

Table 3
Execution of Algorithm 1 for the example given in Fig. 6.

Iter	Q	V	u
1	{}	{3,4,5,6,7,8}	7
2	{7}	{3,4,5,6,8}	3
3	{7,3}	{4,5,6,8}	6
4	{7,3,6}	{4,5,8}	8
5	{7,3,6,8}	{4,5}	5
6	{7,3,6,8,5}	{4}	4
7	{7,3,6,8,5,4}	{}	–

$\mathcal{T}$, i.e., E[$\mathcal{T}$]. Furthermore, we define depth (level) d_i of a node i ($gID = i$) as the length of the path (number of hops) from node i to the root node (base station), and height $\mathcal{H}$ of the MST as the length of the longest path from the root to a leaf node.

Let $\mathcal{T}_i$ be the time when a node i receives the multicast data and $\mathcal{P}_i^j$ be j^{th} ancestor node of node i along the path from node i to the base station, with $\mathcal{P}_i^0 = i$. The value of $\mathcal{T}_i$ can be written as,

$$\mathcal{T}_i = \mathcal{T}_{\mathcal{P}_i^1} + t_i \tag{1}$$

In the above equation, t_i is the time required by node i to receive multicast data provided that its parent has already received the same. After expanding the recurrent equation of $\mathcal{T}_i$, as given in Eq. (1), we get

$$\mathcal{T}_i = \sum_{j=1}^{d_i} t_{\mathcal{P}_i^{j-1}} \tag{2}$$

The value of t_i can be calculated as,

$$t_i = MFD * R_i \tag{3}$$

In the above equation, MFD is the duration of MAC frame and R_i is the number of retransmissions (MAC-frames) required to successfully receive multicast data by node i from its parent. Substituting the value of t_i from Eq. (3) in Eq. (2), we get,

$$\mathcal{T}_i = MFD \sum_{j=1}^{d_i} R_{\mathcal{P}_i^{j-1}} \tag{4}$$

Note that R_i is a geometric random variable with parameter $(1-p)$, where p is the packet loss rate (PLR). In this analysis, we have assumed homogeneous PLR throughout the network. Although this assumption is not true for many practical scenarios, the value of p can be thought of as the PLR of the weakest link in the network, which gives the worst case value of R_i. Furthermore, we assume that the random variables R_i are independent and identically distributed (i.i.d). Let $sumR_i = \sum_{j=1}^{d_i} R_{\mathcal{P}_i^{j-1}}$, substituting the value of $sumR_i$ in Eq. (4), we get

$$\mathcal{T}_i = MFD * sumR_i \tag{5}$$

To calculate the value of $sumR_i$, we use the well-known result which states that the sum of d_i independent and identically distributed geometric random variables with parameter $1-p$ is equivalent to a negative binomial random variable with parameters d_i and $1-p$. Therefore,

$$P(sumR_i = m) = \binom{m-1}{d_i-1}(1-p)^{d_i} p^{m-d_i}, \tag{6}$$

where $m = d_i, d_i+1, d_i+2, \ldots$ The value of $\mathcal{T}$, i.e., the time when a multicast transmission is received by all the nodes in MST, can be approximated by the time when all nodes at level $\mathcal{H}$ have received the multicast transmission. These two definitions for $\mathcal{T}$ are not exactly same, because it could be possible that some node at level less than $\mathcal{H}$ has not received the data, even if all nodes at level $\mathcal{H}$ have received the multicast data. The probability that a node at level less than $\mathcal{H}$ has not received the data given the fact that last node at level $\mathcal{H}$ has received the data, is very less. Moreover, this probability would be even lesser, as we have considered PLR for the poorest link in the network for our analysis. Intuitively, if the farthest nodes with poorest link quality have received the data, then the nodes which are near to the base station with better path quality should have received the data, before the farthest nodes. Moreover, the second definition ensures that all the internal nodes which are in the forwarding path of at least one node at level $\mathcal{H}$, have received the data (Table 4) .

Let Ω be the set of all the nodes in MST which are at level $\mathcal{H}$ and R_{max} be the number of frames elapsed before the last node in Ω receives the multicast transmission, i.e., maximum of $|\Omega|$ random variables distributed by the negative binomial distribution. The value of R_{max} can be written as,

$$R_{max} = max(\{sumR_i : i \in \Omega\}) \tag{7}$$

There is no closed-form formula available for the cumulative distribution function (cdf) of negative binomial distribution, and therefore, computing the expectation E[R_{max}] is not straight forward. We use the approximation results for expectation of maximum of more than one random variables distributed by the negative binomial distribution, from an existing research work given in [10]. According to the results in [10], the value of E[R_{max}] can be written as,

$$E[R_{max}] \approx \log_q|\Omega| + (\mathcal{H}-1)\log_q\log_q|\Omega| \tag{8}$$

In the above equation, $q = 1/p$. Further simplifying Eq. (8), we get

$$\begin{aligned} E[R_{max}] &\approx \frac{\log_e|\Omega|}{log_e q} + (\mathcal{H}-1)log_q\left(\frac{\log_e|\Omega|}{\log_e q}\right) \\ &= \frac{\log_e|\Omega|}{log_e q} + (\mathcal{H}-1)log_q log_e|\Omega| - (\mathcal{H}-1)log_q log_e q \\ &= \frac{1}{\log_e q}(\log_e|\Omega| + (\mathcal{H}-1)log_e log_e|\Omega| \\ &\quad - (\mathcal{H}-1)log_e log_e q) \end{aligned} \tag{9}$$

Next we calculate the value of $|\Omega|$, i.e., the number of nodes in MST at level $\mathcal{H}$. Although the value of $|\Omega|$ highly depends upon node distribution, we consider a WSN with uniform node density and base station at the center. In this case, the number of nodes at level $\mathcal{H}$ can be approximated by the following equation.

$$\begin{aligned} |\Omega| &= \frac{\pi * \mathcal{H}^2 - \pi * (\mathcal{H}-1)^2}{\pi * \mathcal{H}^2} * n \\ &= \frac{2 * \mathcal{H} - 1}{\mathcal{H}^2} * n \le \frac{2}{\mathcal{H}} * n \end{aligned} \tag{10}$$

In the above equation, n is the total number of nodes in MST. Substituting the value of $|\Omega|$ from Eq. (10) in Eq. (9) we get

Table 4
Notations used for the performance analysis of TRM-MAC protocol.

Symbol	Description
$\mathcal{T}$	Time required by all the nodes in MST to receive the multicast transmission
$\mathcal{T}_i$	Time required by node i to receive multicast transmission
t_i	Time required by node i to receive multicast transmission provided that its parent has already received the same
d_i	Length of the path (number of hops) from node i to the root node
$\mathcal{H}$	$max(d_i$: $i \in MST)$, i.e., length of the longest path from the root to a leaf node.
$\mathcal{P}_i^j$	j^{th} ancestor node of node i along the path from node i to the base station, with $\mathcal{P}_i^0 = i$
MFD	MAC frame duration
R_i	Number of retransmissions required by node i, to successfully receive multicast data from its parent.
p	Packet Loss Rate (PLR)
Ω	Set of nodes in MST at level $\mathcal{H}$
f	Reliability (Fraction of nodes which have successfully received the multicast data to the total number of nodes in MST).
$maxRetries$	Number of retransmissions performed by an internal node before discarding the transmission of a multicast data.

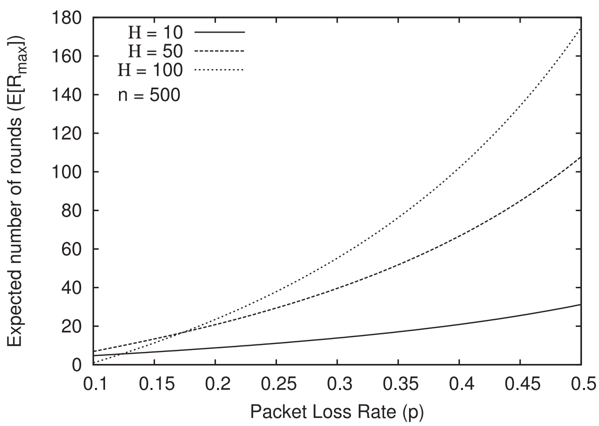

Fig. 8. Expected runtime performance of TRM-MAC with respect to $\mathcal{H}$.

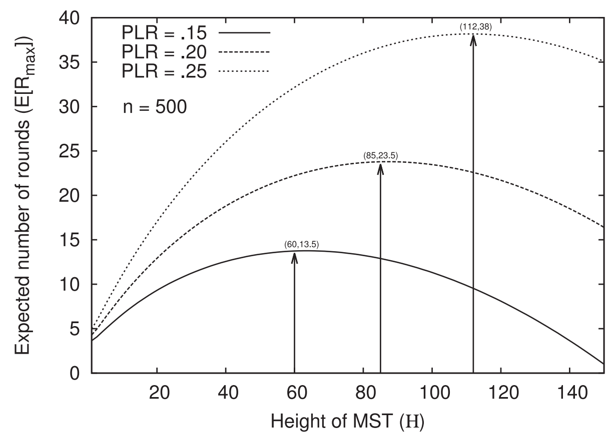

Fig. 9. Expected runtime performance of TRM-MAC with respect to p.

$$E[R_{max}] = \frac{1}{\log_e q}\left(\log_e \frac{2n}{\mathcal{H}} + (\mathcal{H}-1)log_e \log_e \frac{2n}{\mathcal{H}} - (\mathcal{H}-1)log_e log_e q\right) \quad (11)$$

Finally, the value of $E[\mathcal{T}]$ can be calculated as,

$$\begin{aligned} E[\mathcal{T}] &= E[max(\{\mathcal{T}_i : i \in \Omega\})] \\ &= MFD * E[max(\{sumR_i : i \in \Omega\})] = MFD * E[R_{max}] \\ &= \frac{MFD}{\log_e q}\left(\log_e \frac{2n}{\mathcal{H}} + (\mathcal{H}-1)log_e \log_e \frac{2n}{\mathcal{H}} - (\mathcal{H}-1)log_e log_e q\right) \end{aligned} \quad (12)$$

The value of MFD (MAC frame size) depends upon number of internal nodes in MST, the TDMA-scheduling algorithm employed to assign slots in CAP1 and CAP2 portions of the MAC frame, size of multicast data and the underlying data transmission rate.

Fig. 9 shows the variation in $E[R_{max}]$ with respect to p for different $\mathcal{H}$ values keeping the number of nodes in the network as constant, $n = 500$. The value of $E[R_{max}]$ increases with respect to increase in PLR because of higher number of retransmissions at each level of MST. Fig. 8 shows the variation in $E[R_{max}]$ with respect to $\mathcal{H}$. One way to interpret the quantity $\mathcal{H}$ is as the density of the network. For a constant n, small value of $\mathcal{H}$ reflects the higher density networks because, in this case, most of the nodes are nearer to the base station. Initially, the value of $E[R_{max}]$ increases with respect to increase in $\mathcal{H}$. But, after reaching a certain threshold level it starts decreasing further. This is due to the fact that as the network density decreases ($\mathcal{H}$ increases), the number of nodes at level $\mathcal{H}$ ($|\Omega|$) decreases, and therefore, $E[R_{max}]$ starts decreasing. Furthermore, the threshold level after which the $E[R_{max}]$ starts decreasing, increases with respect to increase in PLR. This is because the effect of PLR dominates over the effect of number of nodes in $|\Omega|$ for high PLR values.

6.2. Reliability analysis

We define the reliability of multicast transmission in WSN, as the ratio of the number of sensor nodes that successfully received the multicast data to the total number of sensor nodes in the MST, and denote it by f. The value of f can also be viewed as the probability of multicast data reception at an arbitrary node i in MST. Let Ψ and Φ be the respective events that node i and its parent received the multicast data transmitted by the base station. In that case,

$$\begin{aligned} P(\Psi) &= P(\Psi/\phi) * P(\phi) + P(\Psi/\phi') * P(\phi') \\ &= P(\Psi/\phi) * P(\phi) \quad \because P(\Psi/\phi') = 0 \\ &= (1 - p^{maxRetries}) * P(\phi) \end{aligned} \quad (13)$$

The value of $P(\Psi)$ can be computed recursively in the following manner. If the parent of node i is base station ($d_i = 1$), then $P(\phi) = 1$ as the base station is the source of multicast data. Therefore, the value of $P(\Psi)$ for all the nodes at level 1 is $1 - p^{maxRetries}$. In general, the value of $P(\Psi)$ for a node i will be

$$P(\Psi) = (1 - p^{maxRetries})^{d_i} \quad (14)$$

From Eq. (14), it is clear that the probability of node i receiving the multicast transmission depends upon the level at which it is situated in MST. Higher level implies lesser probability of packet reception. The value of f for an arbitrary node i can be calculated as follows.

$$f = \sum_{l=1}^{\mathcal{H}}((1 - p^{maxRetries})^l * P(d_i = l)) \quad (15)$$

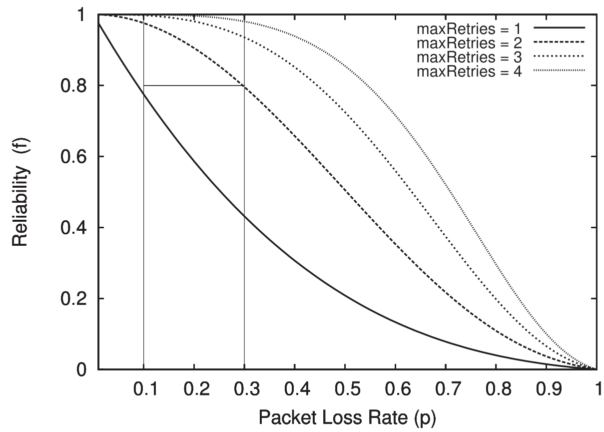

Fig. 10. Reliability performance of TRM-MAC with respect to p for different values of *maxRetries*.

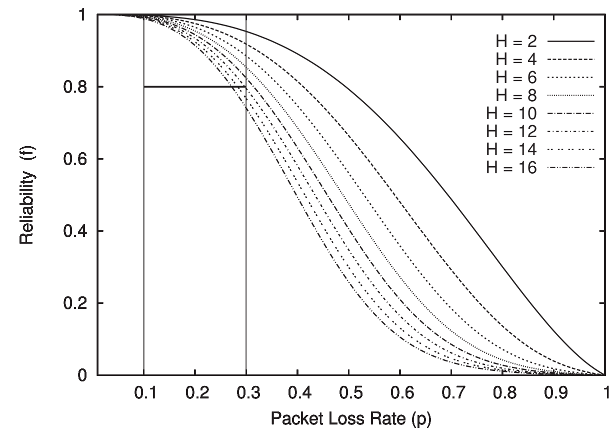

Fig. 11. Reliability performance of TRM-MAC with respect to p for different values of $\mathcal{H}$.

If we consider the WSN with uniform node density and base station at the center, the value of $P(d_i = l)$ can be expressed as the ratio of number of nodes at level l to the total number of nodes in MST. That is,

$$P(d_i = l) = \frac{\pi * l^2 - \pi * (l-1)^2}{\pi * \mathcal{H}^2} = \frac{2 * l - 1}{\mathcal{H}^2} \quad (16)$$

Substituting the value of $P(d_i = l)$ from Eq. (16) in Eq. (15), we get

$$\begin{aligned} f &= \sum_{l=1}^{\mathcal{H}} \left((1 - p^{maxRetries})^l * \frac{2 * l - 1}{\mathcal{H}^2} \right) \\ &= \frac{2}{\mathcal{H}^2} \sum_{l=1}^{\mathcal{H}} \left((1 - p^{maxRetries})^l * l \right) - \frac{1}{\mathcal{H}^2} * \sum_{l=1}^{\mathcal{H}} \left((1 - p^{maxRetries})^l \right) \\ &= \frac{2}{\mathcal{H}^2 * p^{maxRetries}} \left(\mathcal{S} - \mathcal{H} * (1 - p^{maxRetries})^{\mathcal{H}+1} \right) - \frac{1}{\mathcal{H}^2} * \mathcal{S} \\ &= \frac{1}{\mathcal{H}^2} * \mathcal{S} * \left(\frac{2 - p^{maxRetries}}{p^{maxRetries}} \right) - \\ &\quad \frac{2}{\mathcal{H} * p^{maxRetries}} * (1 - p^{maxRetries})^{\mathcal{H}+1}, \end{aligned} \quad (17)$$

where

$$\mathcal{S} = \sum_{l=1}^{\mathcal{H}} \left((1 - p^{maxRetries})^l \right) = \frac{1 - (1 - p^{maxRetries})^{\mathcal{H}+1}}{p^{maxRetries}} - 1 \quad (18)$$

Fig. 10 shows the variation in f with respect to p for different *maxRetries* values keeping $\mathcal{H} = 3$. It is obvious to see that the reliability performance rapidly degrades with the increase in PLR. But, as the number of retries increases, the reliability performance improves considerably. Fig. 10 shows that for practical range of PLR (.1 to .3) for wireless networks, the reliability is more than 80% for $maxRetries > 1$. Similarly, Fig. 11 shows the variation in f with respect to p for different H values keeping $maxRetries = 3$. The reliability performance degrades with the increase in H, which is due to the fact that nodes at longer distant from the base station have lesser probability to receive the multicast data. However, the effect of H on the reliability performance is lesser than that compared to the effect of *maxRetries*. Again, for practical range of PLR, the reliability is more than 80% for $\mathcal{H} < 14$.

7. Simulation results

We have simulated a multihop WSN with 200 nodes randomly distributed over 250 m × 250 m area, using Castalia Simulator [3], to study the performance of TRM-MAC protocol, in terms of reliability, energy and delay parameters. The sensor nodes are based on TelosB hardware platform that uses CC2420 transceiver [4]. The transceivers run at 250 kbps data rate and −3 dbm transmission power which approximately gives 40 m of transmission range, in the absence of any interference. The lognormal shadowing channel model is used to get accurate estimates for average path loss at different receivers. The various values of path loss exponent and gaussian zero-mean random-variable are used to experiment with different PLR values ranging from 1% to 20%. The slot size in CFP1 and CFP2 portion of the MAC frame is taken to be of 1.6 ms and .4 ms respectively, which are sufficiently long for the transmission of DATA frames of size less than 50 bytes and ACK frame of size less than 10 bytes at 250 kbps data rate. The slot size in CAP portion is kept at 128 μs, which is equal to the time required by a node to perform clear channel assessment (CCA).

The reliability is calculated as the ratio of the number of nodes that receive the packet transmitted by the sink node to the total number of nodes in the multicast group. The delay is calculated as the average of time duration from the instant when a packet is sent to the instant when the packet is received at all the group members.

To evaluate the performance of TRM-MAC in terms of energy efficiency, we have used the following simplified energy consumption model.

$$E_{total} = \sum_{i=1}^{n} \left(\sum_{j} (P_j * t_j^i) + \sum E_{trans} \right), \quad (19)$$

where the index j refers to different energy states of the transceiver of a sensor node, namely Sleep, Idle, Transmit and Receive. P_j refers to the power consumed by the transceiver of a sensor node in state j, whereas the term t_j^i refers to the time spent by node i in state j. In order to get more realistic results, we have also considered the amount of energy (E_{trans}) consumed during the transition from one state to another. The value of P_j has been obtained by multiplying the current consumption at −3 dm transmission power and 250 kbps data rate with the supply voltage. The data for current consumption and supply voltage are obtained from the data sheet of transceiver CC2420. Table 5 shows the actual power consumption values for different transceiver states that we have considered in our simulation.

Table 5
Energy consumption of transceiver CC2420 (Data Rate:250 kbps, Transmission Power: −3 dbm).

State	P[mW]	Energy/bit [μJ]
Transmit	46.08	0.105
Receive	56.4	0.128
Idle	1.28	.003
Sleep	.064	≈ 0

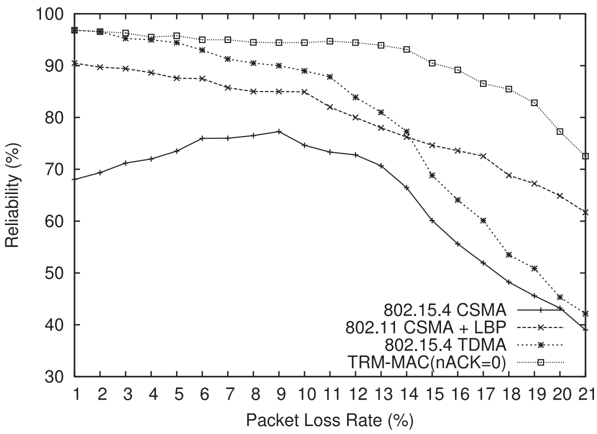

Fig. 12. A comparison of reliability performance between TRM-MAC and other schemes.

The experiment is performed as 1000 simulation runs with random node deployment for each run, to nullify the effect of specific network topology, and each simulation run consist of 1000 packet transmissions. The final results are averaged over all 1000 packet transmissions and 1000 simulation runs. The simulation results for the proposed protocol have been compared with the schemes, which use MST with CSMA-based and TDMA-based MAC respectively, with no MAC layer retransmission, and also with reliable multicast protocol, LBP proposed in [18]. The LBP is the extension of 802.11 MAC protocol [1] for reliable multicast transmission, in which, only a single child node, called leader, transmits the ACK message. The LBP protocol can be seen as the CSMA with retransmission. To simulate the performance of CSMA, we have used IEEE 802.15.4 MAC protocol [2] in non-beacon mode. Similarly, to perform slot assignment for TDMA-based channel access, we have used a Distributed Time Slot Scheduling (DTSS) protocol given in [7]. These protocols, in general, cover most of the existing work discussed in Section 3. To create the MST and perform TDMA scheduling for TRM-MAC, we have used the algorithms given in [34] and [7] respectively.

Fig. 12 shows the reliability performance comparison between TRM-MAC, TDMA, CSMA (IEEE 802.15.4 non-beacon mode) and LBP (an extension of 802.11 to support multicast reliability) channel access protocols, when used along with MST for reliable multicast communication in WSNs. We have kept the value of $maxRetries = 1$, for for each scenario, in the simulation experiment related to Fig. 12. The graph shows that the performance of CSMA (without retransmission) is very poor and this is due to its inability to handle collisions due to correlated-contention caused by simultaneous relaying of multicast data by different internal nodes in a proximity. Furthermore, the problem of collision for CSMA becomes more severe in case of small PLR which implies larger interference range due to lesser path loss. The performance of LBP (CSMA with retransmission) is better than pure CSMA due to its capability to retransmit in case of packet loss. However, it could be possible that some of the child nodes did not get the packet, even if the leader node has received it successfully. In such a situation, the parent node will consider the transmission as successful, but it is actually not. Fig. 12 also shows that the performance of TDMA is better than that of CSMA since it is able to completely avoid the problem of collision. Although there is no retransmission mechanism, the reliability of pure TDMA is limited by the PLR, which becomes unacceptable for higher values of PLR. The TRM-MAC further improves the performance of plain TDMA by incorporating the link layer retransmission by the internal nodes if one of their child has not received the data. Note that, in this experiment, $nACK = 0$ which specifies that the retransmission by an internal node will only happen if one of its *internal-child* has not received the data.

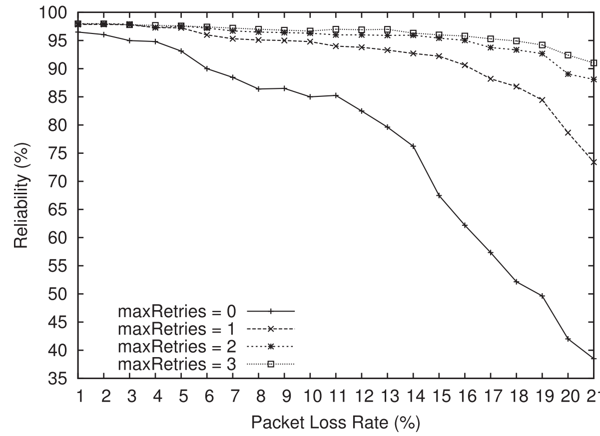

Fig. 13. The effect of parameter maxRetries on the reliability performance of TRM-MAC.

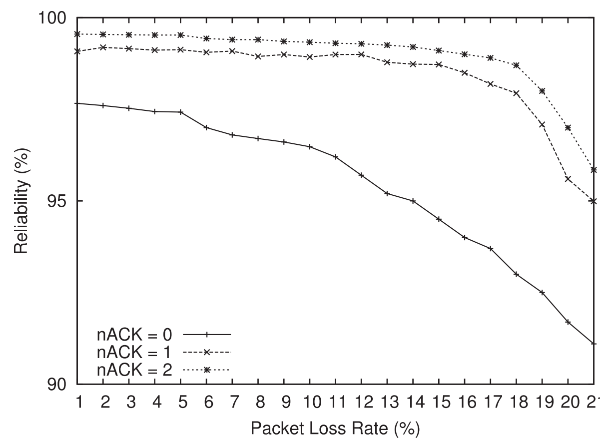

Fig. 14. The effect of parameter *nACK* on the reliability performance of TRM-MAC.

Fig. 13 shows the improvement in reliability performance of TRM-MAC with respect to increase in *maxRetries*, while still keeping $nACK = 0$. It can be seen that the reliability performance starts saturating and does not improve as the value of *maxRetries* increases from 2 to 3. This is due to the fact that, even if a *leaf-child* node has not received the data, it will not transmit any feedback message (ACK or NACK), since $nACK = 0$, and consequently, the internal node will not retransmit the data. Fig. 14 shows the further improvement in reliability performance of TRM-MAC with respect to increase in *nACK*, while keeping $maxretries = 3$. It can be seen that we are able to achieve greater than 99% reliability with $nACK = 1$ for fairly high PLR (<15%). A small fraction (<1%) of failure rate is due to the fact that the *leaf-child* nodes using NACK-based approach may fail to detect a transmission, in case they did not receive the data and any ACK transmitted by the other nodes. At the other extreme, nearly 100% reliability can be achieved by

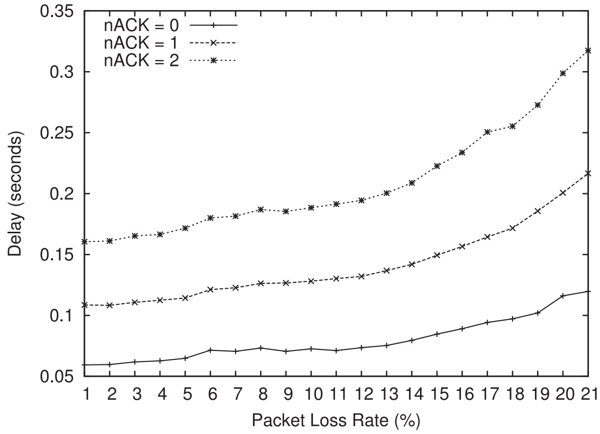

Fig. 15. The effect of parameter *nACK* on the delay performance of TRM-MAC.

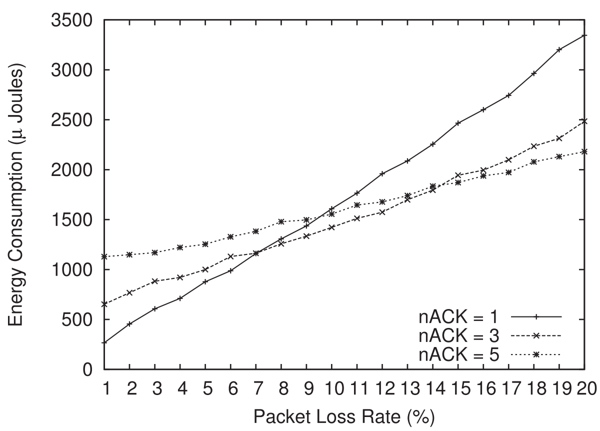

Fig. 16. The effect of parameter *nACK* on the energy consumption performance of TRM-MAC.

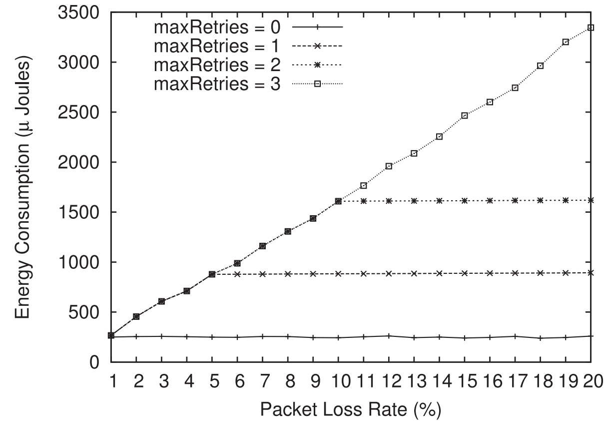

Fig. 17. The effect of parameter *maxRetries* on the energy consumption performance of TRM-MAC.

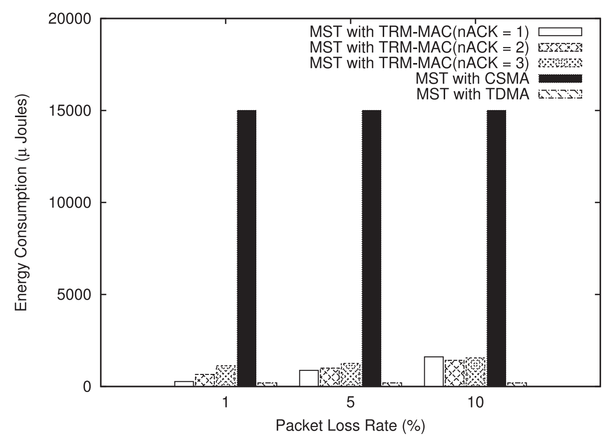

Fig. 18. A comparison of energy performance between TRM-MAC and other similar schemes.

setting *nACK* equal to the total number of children that an internal node can have.

Fig. 15 shows the average delay experienced by the nodes to receive a multicast transmission sent by the sink node, with respect to PLR for various *nACK* values. The delay increases with respect to increase in PLR because of higher number of retransmissions. The rate of increase is even more for larger values of *nACK*. This is because of higher probability of packet loss detection at *leaf-child* nodes, than in case of smaller *nACK* values. The increase in delay by a certain offset, for larger values of *nACK*, is due to larger length of CFP2 portion within the MAC-frame.

Fig. 16 shows the variation in energy consumption of nodes with respect to PLR for various *nACK* values. We have kept the value of $maxRetries = 3$, for each protocol, in the simulation experiment related to Fig. 16. In case of higher PLR, the energy consumption is also high due to a large number of retransmissions of multicast-data by the internal nodes. Additionally, as the PLR increases, the nodes using NACK-approach often have to keep their transceiver active during the CFP2 portion of the MAC-superframe to detect the packet loss by receiving the ACK frames transmitted by the nodes using ACK-approach. However, the effect of *nACK* on the energy consumption is not linear. For small PLR values, many times the leaf nodes using NACK-approach would successfully receive the data transmitted in CFP1 portion of the MAC-superframe, and therefore, they could go to sleep mode immediately after the end of CFP1 portion. In this case, a major portion of energy consumption belongs to the transmission and reception of ACK messages by leaf nodes using ACK-based approach and their parent nodes respectively. This gives the reasoning for increase in energy consumption with respect to increase in *nACK* for small PLR values. As opposed to this, for higher PLR values, as the value of *nACK* increase, the energy consumption decreases. This is due to the decrease in the number of ACK transmissions because, even if the *nACK* value is high, all the nodes using ACK-approach may not receive the data. Secondly, as the number of nodes using NACK-approach would be lesser for higher values of *nACK*, (in case of high PLR) lesser number of NACK-based nodes would spent their energy during CFP2 as well as CAP portion of the MAC-superframe. The above explanation can be summarized as: the rate of increase in energy consumption with respect to PLR is lower for higher values of *nACK*. Finally, the reason for approximately linear increase in energy consumption with respect to increase in PLR, is due to the fact that energy consumption in failed transmissions is lesser as compared to the successful transmissions. Therefore, the energy consumption growth does exactly follow the growth of number of total number of transmissions as PLR increases.

Fig. 17 shows the energy consumption performance of TRM-MAC with respect to increase in *maxRetries*, while keeping $nACK = 1$. It can be seen that after a certain PLR, the energy consumption starts saturating and does not increase further as PLR increases. This is due to the fact that the value of *maxRetries* restricts the actual number of retransmissions at higher PLR, as the internal nodes are not able to successfully forward the data even after a couple of re-transmission attempts.

Finally, Fig. 18 shows the comparison of TRM-MAC with other schemes in terms of average energy consumed by a node. CSMA does not support any sleep and wakeup, and therefore, consumes maximum power. On the other hand, pure TDMA does not support any retransmission scheme, and therefore, the internal nodes can simply sleep immediately after transmission of data at their time slots. The energy consumption of TRM-MAC is in between TDMA and CSMA schemes. Furthermore, the energy consumption is less for smaller values of *nACK* parameter, which can be used to trade-off energy with reliability and delay parameter.

8. Conclusions

In this work, we have suggested a framework for reliable multicast transmission in WSNs and proposed a TDMA-based MAC protocol with hybrid link layer and retransmission scheme within this framework. One of the unique feature of TRM-MAC protocol is its ability to trade-off reliability with delay and energy as per the requirement of applications and channel conditions using *maxRetries* and *nACK* parameters. The simulations and analytical results show that the TRM-MAC protocol is able to considerably improve the reliability performance of multicast communication in WSNs while ensuring reasonably low access delay at the same time. Additionally, a high degree of reliability can be achieved, if a WSN uses TRM-MAC protocol along with an end-to-end reliable multicast solution at link layer and transport layer respectively by appropriately partitioning the task of reliability at both these layers.

References

[1] IEEE standard for information technology telecommunications and information exchange between systems local and metropolitan area network, 2005. IEEE 802.11e standard draft.

[2] IEEE standard for information technology telecommunications and information exchange between systems local and metropolitan area network, 2006. IEEE 802.15.4 standard draft.

[3] Castalia network simulator, 2016, Available at https://castalia.forge.nicta.com.au/index.php/en/documentation.html, Accessed on April 2.

[4] CC2420 data sheet, 2016, https://inst.eecs.berkeley.edu/~cs150/Documents/CC2420.pdf, Accessed on April 2.

[5] A. Bhatia, R.C. Hansdah, A TDMA-based energy aware MAC (TEA-MAC) protocol for reliable multicast in WSNs, in: 2014 IEEE 28th International Conference on Advanced Information Networking and Applications (AINA), 2014, pp. 798–805.

[6] A. Bhatia, R.C. Hansdah, RD-TDMA: a randomized distributed TDMA scheduling for correlated contention in WSNs, in: Proceedings of International Conference on Advanced Information Networking and Applications Workshops (WAINA), 2014, 2014, pp. 378–384.

[7] A. Bhatia, R.C. Hansdah, A distributed TDMA slot scheduling algorithm for spatially correlated contention in WSNs, in: Proceedings of 27th International Conference on Advanced Information Networking and Applications Workshops, 2013, pp. 377–384.

[8] G. Chelius, E. Fleury, NP-completeness of ad hoc multicast routing problems, 2016, Technical Report RR-5665, Available at http://hal.inria.fr/inria-00070343, Accessed on April 2.

[9] R. Flury, R. Wattenhofer, Routing, anycast, and multicast for mesh and sensor networks, in: Proceedings of INFOCOM, IEEE, 2007, pp. 946–954.

[10] P.J. Grabner, H. Prodinger, Maximum statistics of n random variables distributed by the negative binomial distribution, Comb., Probab. Comput. 6 (6) (1997) 179–183.

[11] D.S. Hochbaum, Approximation Algorithms for NP-Hard Problems, in: Approximating Covering and Packing Problems: Set Cover, Vertex Cover, Independent Set, and Related Problems, PWS Publishing Co., 1997, pp. 94–143.

[12] P. Huang, L. Xiao, S. Soltani, M.W. Mutka, N. Xi, The evolution of MAC protocols in wireless sensor networks: a survey, IEEE Commun. Surv. Tut. 15 (1) (2013) 101–120.First

[13] J.G. Jetcheva, D.B. Johnson, Adaptive demand-driven multicast routing in multi-hop wireless ad hoc networks, in: Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking & Computing, 2001, pp. 33–44.

[14] H. Karl, A. Willig, Protocols and Architectures for Wireless Sensor Networks, in: Transport Layer and quality of service, John Wiley & Sons, 2005, pp. 376–377.

[15] D. Koutsonikolas, S. Das, H.Y. Charlie, I. Stojmenovic, Hierarchical geographic multicast routing for wireless sensor networks, in: Proceedings of International Conference on Sensor Technologies and Applications (SensorComm '07)., 2007, pp. 347–354.

[16] S.S. Kulkarni, M. Arumugam, Infuse: A TDMA based data dissemination protocol for sensor networks, Int. J. Distr. Sensor Netw. 2 (1) (2006) 55–78.

[17] S. Kumar, S. Chauhan, Article: a survey on scheduling algorithms for wireless sensor networks, Int. J. Comput. Appl. 20 (5) (2011) 7–13.

[18] J. Kuri, S.K. Kasera, Reliable multicast in multi-access wireless lans, Wireless Netw. 7 (4) (2001) 359–369.

[19] A. Okura, T. Ihara, A. Miura, Branch aggregation multicast (BAM): An energy efficient and highly compatible multicast protocol for wireless sensor networks, IEICE - Trans. Inf. Syst. (2006) 1633–1643.E89-D(5)

[20] S.-J. Park, R. Sivakumar, I.F. Akyildiz, R. Vedantham, GARUDA: achieving effective reliability for downstream communication in wireless sensor networks, IEEE Trans. Mobile Comput. 7 (2) (2008) 214–230.

[21] J. Peng, A new ARQ scheme for reliable broadcasting in wireless LANs, IEEE Commun. Lett. 12 (2) (2008) 146–148.

[22] S. Ramanathan, A unified framework and algorithm for (t/f/c)DMA channel assignment in wireless networks, in: Proceedings of INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution, 2, 1997, pp. 900–907.

[23] B.R. Chen, K.K. Muniswamy-reddy, M. Welsh, Ad-hoc multicast routing on resource-limited sensor nodes, in: Proceedings of the 2nd International Workshop on Multi-hop Ad Hoc Networks: from Theory to Reality, ACM Press, 2006, pp. 87–94.

[24] C. Shanti, A. Sahoo, Treefp: A tdma-based reliable and energy efficient flooding protocol for wsns, in: Proceedings of the 2011 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, WOWMOM '11, IEEE Computer Society, 2011, pp. 1–7.

[25] A. Sheth, B. Shucker, R. Han, VLM2: a very lightweight mobile multicast system for wireless sensor networks, in: Proceedings of IEEE Wireless Communications and Networking Conference (WCNC), 2003, pp. 1936–1941.

[26] R. Silva, T. Camilo, A. Rodrigues, M. Silva, F. Gaudencio, F. Boavida, Multicast in wireless sensor networks the next step, in: Proceedings of 2nd International Symposium on Wireless Pervasive Computing (ISWPC '07'), 2007.

[27] F. Stann, J. Heidemann, Rmst: Reliable data transport in sensor networks, in: Proceedings of the First International Workshop on Sensor Network Protocols and Applications, 2003, pp. 102–112.

[28] M. Sun, L. Huang, A. Arora, T.h. Lai, Reliable mac layer multicast in IEEE 802.11 wireless networks, in: Proceedings of ICPP02, 2002, pp. 527–536.

[29] B. Sundararaman, U. Buy, A.D. Kshemkalyani, Clock synchronization for wireless sensor networks: a survey, Ad Hoc Netw. (Elsevier) 3 (2005) 281–323.

[30] A.R. Swain, R.C. Hansdah, A model for the classification and survey of clock synchronization protocols in WSNs, Ad Hoc Netw. 27 (2015) 219–241.(C)

[31] K. Tang, M. Gerla, MAC reliable broadcast in ad hoc networks, in: Proceedings of Military Communications Conference, 2001. MILCOM., 2, 2001, pp. 1008–1013.

[32] T.Y.T. Niwat, S. Kaoru, An efficient multicast routing protocol for wireless sensor networks, Inst. Electr., Inf. Commun. Eng., IEIC 104 (2005) 419–422.

[33] C.Y. Wan, A.T. Campbell, L. Krishnamurthy, Pump-slowly, fetch-quickly (PSFQ): a reliable transport protocol for sensor networks, IEEE J. Select. Areas Commun. 23 (4) (2005) 862–872.

[34] Q. Ye, L. Cheng, A lightweight approach to mobile multicasting in wireless sensor networks, Int. J. Ad Hoc Ubiquitous Comput. 2 (1/2) (2007) 36–45.

[35] L. Xiao, G. Zeng, C. Wang, Grid Multicast: an Energy-Efficient Multicast Algorithm for Wireless Sensor Networks, in: Proceedings of Fourth International Conference on Networked Sensing Systems, 2007. INSS'07, IEEE, 2007, pp. 267–274.

Ashutosh Bhatia received his B.E. degree in Computer Science from Barkatullah University, Bhopal, India, in 2000. From 2001 to 2005 he worked as a scientist in Defence Research and Development Organization (DRDO), India. He received his M.E. degree in Computer Science from Indian Institute of Science (IISc), Bangalore, India, in 2007. From 2007 to 2010, he worked as research engineer in Samsung India Software Operations (SISO). Currently he is a Ph.D. candidate in the Department of Computer Science and Automation (CSA) at Indian Institute of Science (IISc), Bangalore, India. His research interests include building new designs, protocols, algorithms and theories that improve the robustness and performance of computer networks.

R. C. Hansdah received the B.E. degree in Electronics and Communication Engineering, and the M.E. degree in Automation from Indian Institute of Science, Bangalore, India, in the year 1980, and 1982, respectively. He received the Ph.D. degree in Computer Science from Indian Institute of Science, Bangalore, India, in 1989. He joined the Department of Computer Science and Automation, Indian Institute of Science, Bangalore, India as a Lecturer in October 1989. Since then he has been a faculty in the same Department, and currently he is an Associate Professor. His research interests are in the area of Real-Time Operating Systems, Computer Networks, and Distributed Systems Security.