2009

# A novel real-time MAC layer protocol for wireless sensor network applications

B.K. Singh

Kemal Tepe
*University of Windsor*

# A Novel Real-time MAC Layer Protocol for Wireless Sensor Network Applications

Brajendra Kumar Singh
Department of Electrical and Computer Engineering
University of Windsor
Windsor, Ontario
Email: singh11g@uwindsor

Kemal Ertugrul Tepe
Department of Electrical and Computer Engineering
University of Windsor
Windsor, Ontario
Email: ktepe@uwindsor.ca

*Abstract*—**This paper presents a comparative study of existing real-time MAC layer protocols for wireless sensor networks. Then, a new real-Time MAC protocol is presented that is based on a general purpose MAC protocol, called S-MAC. While medium access strategy in S-MAC is based on contention and back-off schemes, protocol proposed in this paper uses feedback approach as a medium access strategy. As a result of this, it increases consistency in data transmission pattern, which enables it to guarantee end-to-end delay deadlines for soft real-time applications. Proposed protocol works in continuous ON mode of operation at MAC layer and is intended to be used for randomly deployed single stream wireless sensor applications. Finally, a comparative performance analysis of proposed real-time protocol is done with other real-time and general purpose MAC protocols for wireless sensor networks.**

## I. INTRODUCTION

A wireless sensor node is highly energy constrained device and often it's not feasible to change batteries of these nodes after their deployment in the field. Therefore, most of research in wireless sensor network (WSN) focuses on energy conservation. However, with the advancement of communication and MEMS technologies, wireless sensor network are increasingly used for time critical applications. In addition to this, there are applications in which energy consumption assumes secondary importance such as sensor based natural disaster management systems. For example, in case of forest fire or flood situation, a WSN is randomly deployed in affected area and it should monitor events continuously on real-time basis for the duration of calamity. Thus, it does not matter even if wireless sensor nodes die after this duration.

For a given sensor hardware, role of a real-time MAC layer protocol is extremely important for any time critical development at higher layers. A real-time MAC layer protocol should guaranty bounded end-to-end delay. End-to-end delay should be as small as possible. This paper surveys available real-time MAC layer protocols for WSNs. There are seven real-time MAC layer protocols available in the literature for WSNs. Thus, in this paper, we make an effort to carry out a comparative study of these protocols, prior to proposing our real-time MAC protocol.

Section II of this paper presents a comparative study of existing real-time MAC protocol. Problem analysis is done in Section III. Section IV presents description of proposed protocol. Information about simulation is given in Section V. Section VI presents results and related discussion. Conclusion and future work are presented in section VII.

## II. RELATED WORK

There are in general two types of MAC protocols. These are random access protocols such as CSMA based protocols and deterministic scheduling protocols such as TDMA based protocols. In general, it is relatively easier to define delay deadline at MAC layer with deterministic scheduling protocols. However, TDMA based protocols suffer many disadvantages with regard to real-time application requirements at MAC layer in WSNs. For example, TDMA based MAC schemes can not adapt well to frequently changing load condition, thus, they are not good for event driven reporting. Though TDMA based protocol guaranty a bounded end-to-end packet delay, but in general, these delay bounds are very high compared to contention based protocol.

There are several general purpose contention based MAC layer protocols proposed in the literature. For example, S-MAC [1] is a widely referred general purpose MAC protocol for WSNs that has fix duty cycle. More information about S-MAC with respect to real-time communication is presented in Section III. T-MAC [2] is also a widely referred MAC protocol for WSNs. T-MAC supports adaptive duty cycle mode of operation that facilitates change in duty cycle during run time as per changing load conditions. T-MAC itself is not a good candidate for real-time communication due to its inconsistent data transmission pattern as in T-MAC, all data packets keep on traveling together toward the sink like a bunch, which means that the first packet will reach to the sink very late, but subsequent data packets will follow the first packet shortly. D-MAC [3,4] is another widely referred low sleep latency MAC protocol for unidirectional source to sink communication pattern in tree topology WSNs in which sink node is placed at the top of a tree. Philosophy of D-MAC protocol is not suitable for randomly deployed real-time WSN applications due to need for global synchronization and scalability problem. Additionally, it takes time for nodes to come up with a new staggered sleep and wake up schedule for whole network due a single node failure in tree topology network. Though, S-MAC, and D-MAC are not real-time protocols per se, but they are included because some of real-time MAC protocols explained later in this section are based on these protocols. Our study concluded that there are seven real-time MAC protocols available in the literature. These are VTS, I-EDF, Dual-mode

978-1-4244-3355-1/09/$25.00©2009 IEEE

real-time MAC protocol, TOMAC, PR-MAC, CR-SLF and RRMAC. These protocols are discussed here in the following paragraphs of this section.

Virtual TDMA for Sensors (VTS) protocol [5,6] is designed for soft real-time WSN applications. VTS is based on S-MAC protocol. In VTS, packet travels strictly one hop per TDMA slot (which is a frame in S-MAC protocol) as per its virtual TDMA arrangement since only one node can transmit per slot. VTS has few problems with regard to real-time communication. First, as it is not possible to facilitate packet transfer to several hops in a given slot duration in VTS, thus, it can not provide speed up to alarm messages by varying duty cycle of a TDMA slot. Second, in VTS, number of slots in a TDMA frame equals to number of nodes in range. As any node gets a transmission slot again after $m$ slots in a TDMA frame with $m$ nodes, thus it puts a limit to packet arrival interval ($T_{AI}$) at MAC layer in the source node for transmission. Precisely, $T_{AI}$ can be $mT_s$, $2mT_s$, $3mT_s$,…. etc., where $T_s$ represents slot duration in a TDMA frame in VTS. Thus, VTS can not work for WSN applications that has higher packet generation rate (i.e. $T_{AI} < mT_s$) for a given slot duration. In general, it can be said that though VTS gives timeliness guarantees, but these guarantees are too large, which makes it a quite slow protocol as compared to contention based protocols (such as S-MAC, T-MAC and our proposed real-time protocol) with ON, and frame duration of these contention based protocols equal to ON and slot duration respectively of TDMA frame of VTS. Third, being TDMA based protocol, VTS does not maximize spatial channel reuse.

Implicit Earliest Deadline First (I-EDF) [7,8] algorithm schedules message for transmission with smaller deadline first at MAC layer. It is a hard real time MAC layer protocol for WSNs. This protocol also has some problems with regard to real-time communication. First, it is designed for periodic data transmission and is not suitable for event driven WSN applications. Second, it assumes cellular network structure and is not suitable for randomly deployed WSNs. Third, it uses specialized hardware such as more capable sensor nodes called routers for cell to cell communication. Fourth, I-EDF needs multi-channel radio sensor hardware. Fifth, I-EDF needs synchronization on global basis.

Dual-mode real-time MAC protocol [9,10] is a hard real-time MAC protocol and is based on I-EDF protocol. It has two modes. In the protected mode, message travels slowly but reliably, while in unprotected mode, the message travels with full speed but unreliably. On comparing Dual-mode real-time MAC protocol with our proposed real-time protocol, it is observed that both protocols are closely related in certain aspects. For example, both protocols are meant for randomly deployed WSN, and both avoid collision by stopping neighboring nodes from initiating a transmission for certain duration. However, there are many differences between Dual-mode real-time MAC protocol and our real-time protocol. First, Dual-mode real-time MAC protocol uses reservation mechanism to avoid collision, while our protocol uses feedback mechanism. Second, Dual-mode real-time MAC protocol relies heavily on global synchronization mechanism. It requires that all nodes need to know their absolute position information. This calls for special requirements such as use of GPS-enabled moving vehicle to inform nodes about their absolute positions after deployment. Such arrangement is hard to achieve for a randomly deployed WSNs. In contrast to this, real-time MAC protocol proposed in this paper regulates medium access based on relative position of sensor nodes. In addition to this, energy consumption behavior or fault tolerance mechanism of Dual-mode real-time MAC protocol are not provided in [9,10].

TOMAC protocol [11] provides hard real-time message ordering at MAC layer using non destructive bit wise arbitration for one hop mesh network. This protocol is hard to generalize for multi-hop network and other communication topologies.

Path Oriented Real-time MAC (PR-MAC) [12] is a soft real-time protocol for tree based WSNs. It is based on D-MAC protocol and has staggered sleep and wakeup schedule with respect to sink node. Additionally, PR-MAC assumes multi-channel radio. PR-MAC has two normally ON (listen) durations in a frame (work cycle), consequently, it facilitates bidirectional end-to-end packet transfer in one frame duration. PR-MAC targets persistent periodic WSN applications, where communication path remains unchanged for quite some time and is not suitable for event driven real-time WSN applications. Adequate fault tolerance is needed in PR-MAC to avoid deadlock due to possible repeated collision between an out of sync node and its neighbor. Being tree based protocol, PR-MAC needs global synchronization, and has no spatial channel reutilization.

Channel Reuse-based Smallest Latest-start-time First (CR-SLF) [13] algorithm schedules messages at MAC layer to increase spatial channel reuse in soft real-time multi-hop WSNs. CR-SLF is developed for mobile wireless sensor network such as a network of mobile robots with each robot having a wireless sensor device attached to it. CR-SLF uses centralized scheduling algorithm, in which a centralized scheduler decides as to when and who will transmit or receive messages. Being centralized algorithm, CR-SLF is not scalable. It also needs up-to-date global position information of mobile wireless sensor nodes prior to scheduling or medium access decision.

RRMAC [14] is a TDMA based hard real-time MAC protocol for a multi-hop convergecast WSN. Its TDMA superframe assigns time slots to sensor nodes in a hierarchical tree structure with base station at the top of a tree. Sensor nodes form clusters with one cluster head for each cluster. Thus, data from sensor nodes in a cluster reaches to the top of a tree in one super frame. Hierarchical time slot assignment is similar to staggered schedule of D-MAC [3,4] or PR-MAC [12]. However, unlike PR-MAC or D-MAC, RR-MAC super frame is based on IEEE 802.15.4 frame structure and only upper level cluster heads (not the lower level sensor nodes) have dedicated time slots in the TDMA superframe. Cluster head aggregates data collected from lower level sensor nodes and forwards them to upper cluster head in hierarchy. RRMAC assumes that sensor nodes have two RF power levels, thus, cluster head are more powerful node in terms of their transmission range as compared to other sensor nodes in lower level of hierarchy. Being tree based protocol, RRMAC has scalability issues due to constraint of superframe length and amount of data aggregation possible. In large randomly deployed multi-hop WSN, maintaining global synchronization is difficult.

## III. PROBLEM ANALYSIS

S-MAC is taken as basis for our protocol development because it is a general purpose, widely referred contention based MAC protocol. Its source code and results are easily available in literature. S-MAC can work with or without periodic sleeping. However, Reference [1] mentions that continuous ON mode of operation in S-MAC is intended mainly for protocol evaluation purpose. S-MAC uses contention and back-off schemes for wireless medium access. It also uses RTS/CTS exchange method to avoid hidden terminal problem of wireless communication. However, in general, it's not known a priori as to which wireless sensor node will win contention in a neighborhood. Therefore, it's not possible to predict about data transmission pattern in the network. For example, Figure 1 shows one of such possible data packet transfer pattern in S-MAC protocol. In this figure, if node $N_0$ has four packets to send to sink node $N_{10}$, then $N_0$ starts contending for medium at $t_1$ and wins contention. Thus, $N_0$ send first packet P0 to $N_1$ node. At $t_2$, both $N_0$ and $N_1$ contend for medium. Now, if $N_0$ again wins contention, then it sends second packet P1 to $N_1$. However, from real-time point of view, it is desired that as P0 appears first in the network (which could be an alarm for an event), then node $N_1$ should have won contention and should have forwarded P0 to $N_2$ node, instead of loosing contention to $N_0$ node and receiving next packet P1. As we see between time $t_3$ to $t_7$, it's totally unpredictable as to which node is wining contention. To remove this uncertainty, we introduced a feedback control packet, called Clear Channel (CC), in S-MAC. Thus, medium access strategy in our proposed real-time MAC protocol is based on CC control packet. It is explained in next section.

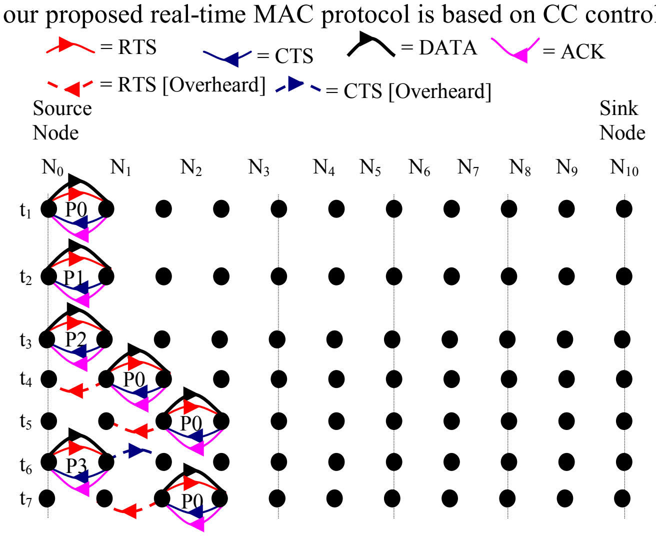


Figure 1. A possible data packet transfer pattern in S-MAC protocol

## IV. PROTOCOL DESCRIPTION

The proposed real-time MAC protocol is for single stream communication. It means that there is only one source and one sink during lifetime of a communication stream in a randomly deployed WSN. It assumes that transmitting power of a sensor node remains same during run time of WSN. Additionally, transmitting power of node is set during initialization phase of protocol in such a way that a node could reach to its one hop neighbor only. During initialization phase of protocol, if there is a wireless sensor node between two reachable wireless sensor nodes with typical transmitting power, then the middle node goes into sleep mode. This assumption in our protocol is inherited from S-MAC. Additionally, unlike S-MAC, all control packets have same contention duration in proposed real-time MAC protocol.

Working of proposed real-time MAC protocol is based on use of CC control packet. CC is used to assign an appropriate value to Clear Channel Flag (CCF) of every sensor node. CCF is a Boolean variable. Central idea of this protocol is that the node that has CCF as 1 can transmit as well as receive data packets, while it can only receive if its CCF value is 0. Initially all nodes have CCF value as 1. CC control packet have a Clear Channel Counter (CCC), which is an integer variable. Its value ranges from 0 to 3. The value of CCC is 3 at the originating node of CC and decreased by one with one hop transmission of CC. CC is always transmitted from sink to source direction. If value of CCC of CC control packet is 2 or 3 in a node, then CCF of that node will remain 0, which means that it can not initiate a data packet transmission. However, if value of CCC of CC is 0 or 1 in a node, then CCF of that node will become 1, which means that it can now initiate a data packet transmission.

Figure 2 explains working of proposed protocol. In this figure, $N_0$ is a source node and $N_{10}$ is a sink node. In the beginning, $N_0$ has several packets (P0, P1, …etc.) to send to sink node. Now, $N_0$ sends RTS to $N_1$. $N_1$ responds with CTS. Then, $N_0$ sends P0 to $N_1$. After receiving P0, $N_1$ sends ACK to $N_0$. This completes one data transfer cycle by one hop. Duration of one data transfer cycle is designated by *Tx* and duration of one control packet is designated by *Tc*. After getting ACK, $N_0$ sets its CCF value to 0, which means that it can not transmit next packet unless its CCF value is set to 1 again. Now, in second *Tx* duration, $N_1$ sends P0 to $N_2$ and sets its own CCF value to 0. In third *Tx* duration, $N_2$ sends P0 to $N_3$ and sets its own CCF value to 0. In fourth *Tx* duration, $N_3$ sends P0 to $N_4$ and sets its own CCF value to 0. Each packet has a Hop Counter (HC) integer variable whose value varies from 0 to 4 for first 4 hops of a communication stream and 0 to 2 for all later 2 hops segments of the communication stream. At $N_0$, the value of HC of P0 is 4 and it is decreased by one each time P0 is transmitted successfully by one hop. Once P0 reaches to $N_4$ node, its HC becomes 0. Then, $N_4$ sets HC of P0 to 2, which will become 0 again after next 2 hops transmission. Now, after successful transmission of ACK to

$N_3$, $N_4$ waits for 2*Tc* duration prior to forwarding P0 to $N_5$. Meantime, in the first *Tc* duration after receiving ACK from $N_4$, N3 sends CC signal to $N_2$ and sets its CCF to 0. In second *Tc* duration, $N_2$ sends CC signal to $N_1$ and sets its CCF to 0. In third Tc duration, $N_1$ sends CC signal to $N_0$ and sets its CCF to 1. Thus, after getting CC from $N_1$ node, $N_0$ can transmit new packet P1 to $N_1$ in next one *Tx* duration and after that, $N_1$ can also forward P1 to $N_2$ in second *Tx* duration and wait there for next CC control packet.

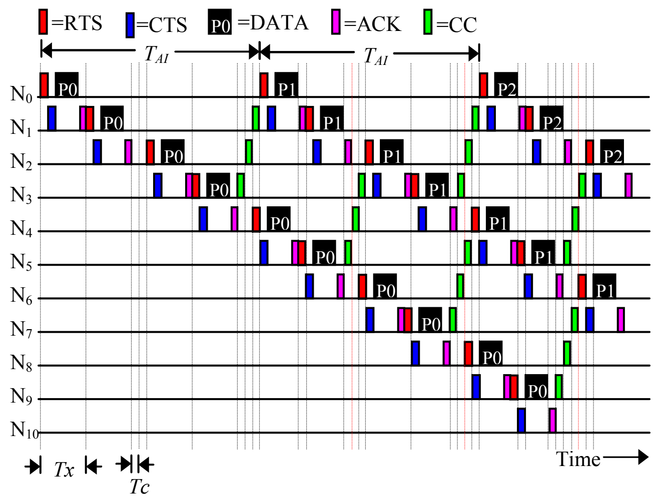


Figure 2. Timing diagram of packet transfer in proposed protocol

*End-to-end delay bounds for data transfer in continuous mode of operation of MAC:*

We calculate these bounds for worst case load scenario, in which all packets are available for transmission at time t = 0. As shown in Figure 2, data transfer duration by one hop is denoted by *Tx*. The control packets RTS, CTS, ACK and CC have the same interval that is denoted by *Tc*. Here, *n* is number of hops from the source to the destination, which signifies that there are *n*+1 nodes in the network.

Thus, as per Figure 2, data transfer duration by one hop is as follows.

$$Tx = \sum \{\text{duration of (RTS+CTS+DATA+ACK)}\}$$

$$Tx = Tc + Tc + Td + Tc$$

$$Tx = Td + 3\ Tc \qquad (1)$$

Time taken by $m_{th}$ packet to reach from the source to the destination in *n* hops is denoted at $T(m,n)$. Thus, time taken by the first packet to reach the destination node is given as $T(1,n)$:

*for even n:*

$$T(1,n) = Tx + Tx + Tx + Tx + 2\ Tc + Tx + Tx + 2\ Tc \ldots\ldots + 2\ Tc + Tx + Tx \quad \text{upto } n$$

$$T(1,n) = 4Tx + \sum_{i=1}^{\left(\frac{n-4}{2}\right)} (2Tc + 2Tx)$$

$$T(1) = n\ Tx + (n\text{-}4)\ Tc$$

Substituting *T*x from (1)

$$T(1,n) = n\ Td + 4(n\text{-}1)\ Tc \qquad (2)$$

The start of second packet is delayed by an offset (represented as $T_{AI}$ in Figure 2) given as below:

$$\text{Offset} = Tx + Tx + Tx + Tx + Tc + Tc + Tc$$

$$\text{Offset} = 4\ Tx + 5\ Tc$$

Substituting *T*x from (1)

$$\text{Offset} = 4\ Td + 17\ Tc \qquad (3)$$

Then time taken by second packet to reach the destination node is denoted as $T(2,n)$ and is given below:

$T(2,n)$ = [offset] + [time taken by a packet to travel *n* hops]

From (2) and (3)

$$T(2,n) = [4\ Td + 17\ Tc] + [n\ Td + 4(n\text{-}1)\ Tc] \qquad (4)$$

Similarly, time taken by $m_{th}$ packet to reach the destination node, denoted as $T(m,n)$, and is given below:

$T(m,n)$ = ($m$-1)*[offset] + [time taken by a packet to travel *n* hops]

Here * represent multiplication operation.

$$T(m,n) = (m\text{-}1)[4\ Td + 17\ Tc] + [n\ Td + 4(n\text{-}1)\ Tc]$$

$$T(m,n) = (4m+n\text{-}4)\ Td + (17m+4n\text{-}21)\ Tc \qquad (5)$$

Following the same method as above, expression for odd n is given below:

$$T(m,n) = (4m+n\text{-}4)\ Td + (15m+4n\text{-}22)\ Tc \qquad (6)$$

By observing equation (5) and (6), it can be said that end-to-end data transfer delay will be more or less remain same irrespective of fact that total number of hops are odd or even.

## V. SIMULATION

We used Omnet++ simulator for our simulation. We used minimum hop routing for all protocol taken for simulation study. Major parameter values taken for our simulation study are given in TABLE I. Here, 1 Tic of a crystal oscillator of sensor node is taken as 30.518 µSec. Parameter values taken in our simulation are based on S-MAC [1], T-MAC [2] and VTS [5,6] papers.

TABLE I. PARAMETER VALUES IN SIMULAION

| Parameter | Value |
|---|---|
| Power consumption in receive or idle listening mode | 14.4 mW |
| Power consumption in transmit mode | 36 mW |
| Power consumption in sleep mode | 15 µW |
| Radio bandwidth | 40 Kbps |

| | |
|---|---|
| Maximum value of RTS contention time without backoff (in S-MAC) | 300 Tics |
| RTS contention time (in T-MAC) | 300 Tics |
| Frame duration | 20000 Tics |
| Typical value of RTS,CTS,ACK, CC durations in proposed real-time MAC protocol | 47 Tics |
| Typical value of RTS in VTS | 47 Tics |
| Typical value of CTS, and ACK duration in S-MAC, T-MAC and VTS | 47 Tics |
| DATA duration in S-MAC, T-MAC, VTS and proposed real-time MAC protocol | 132 Tics |
| Typical value of one data transfer cycle duration in S-MAC and T-MAC | 450 Tics |
| Typical value of one data transfer cycle duration in VTS and proposed real-time MAC protocol | 275 Tics |
| Data packet size | 20 Byte |
| Average VTS super frame length | 20 slots |
| VTS slot duration | 20000 Tics |
| Session duration for proposed real-time MAC protocol, SMAC and T-MAC | 25 Sec |
| Session duration for VTS | Till end of data transfer |

## VI. RESULTS AND DISCUSSION

The real-time protocol proposed in this paper operates with continuous ON mode of operation, while S-MAC can operate both with and without duty cycle. Therefore, results for our protocol and S-MAC are for continuous ON mode of operation of wireless sensor node. However, T-MAC and VTS are operated for 99% duty cycle as continuous ON mode of operation is not possible in these protocols. Operating T-MAC and VTS at 99% duty cycle means that that wireless sensor nodes in these protocols are not transmitting any data packets for one SYNC duration at the beginning of each frame and its ON duration is also less by one percent as compared to proposed real-time MAC protocol. Thus, for a frame size of 20000 tics, T-MAC and VTS are operating approximately 352 tics less as SYNC duration is typically 152 tics and 1% of frame duration is sleep time (i.e. 200 tics in this case). Its makes a difference of approximately one data packet transmission per frame as one data packet takes typically around 273 tics in our simulation. Thus, results for 100% ON time and 99% ON time do not differ significantly and give us a fair comparison of performance parameters of these protocols.

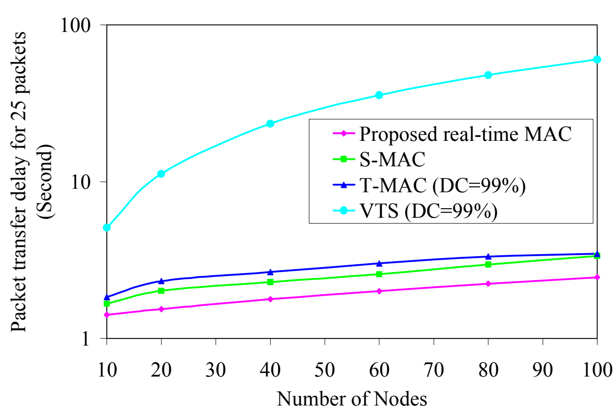


Figure 3. Packet transfer delay pattern for 25 packets

Packet transfer delay is calculated as duration between start of the first packet transfer at the source to the time of successful reception of last packets at the destination node. As shown in Figure 3, packet transfer delay for 25 packets is less in proposed real-time MAC protocol as compared to S-MAC, T-MAC and VTS protocol. It is due to fact that our protocol uses feedback mechanism that reduces contention duration and collision significantly. VTS protocol has highest packet transfer delay due to fact that packet transfer in VTS is done as per TDMA philosophy. In VTS, a time slot in super frame is equal to frame duration of S-MAC protocol. Thus, in one frame duration, all packets can be transmitted by one hop only, while, in other three protocols compared here, packets can travel by several hops in a given frame duration.

Packet overhead is calculated on basis of average number of control packets needed for one data packet transfer by one hop. Control overhead packets in proposed real-time MAC protocol are RTS, CTS, ACK and CC. Figure 4 shows that packet overhead is highest for our protocol. It is due to fact that our protocol uses an extra CC control packet. Packet overhead is lowest (close of three) in case of VTS. It is due to fact that VTS is a virtual TDMA based collision free protocol. However, as VTS still uses S-MAC (contention based) as underlying protocol for data packet transfer, thus, it has packet overhead close to 3 due to RTS, CTS and ACK control packets for a data packet transmission by one hop. Here, is observed that though packet overhead is more in proposed real-time MAC protocol, still end-to-end packet transfer delay, as shown in Figure 3, is lowest on our protocol. It is due to fact that our protocol reduces contention duration and collision due to feedback based medium access approach. Thus, gain in terms of time saved due to less contention duration and less collision is more than time taken by extra CC control packets.

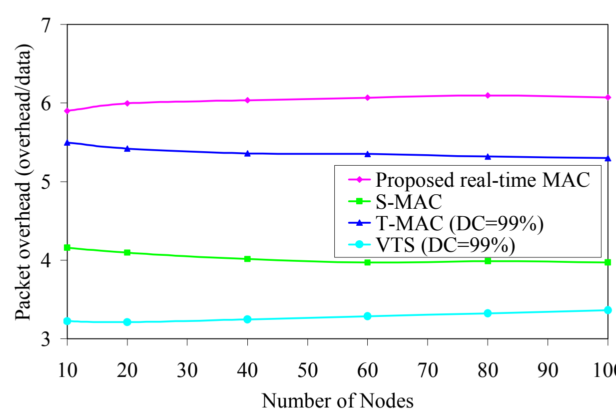


Figure 4. Packet overhead pattern

Energy consumption includes energy spent by all nodes in network during transmission, reception, idle listening and sleep mode. Parameter values taken for energy consumption are mentioned in TABLE I. We observed energy consumption behavior S-MAC, T-MAC and our protocol for session duration of 25 seconds. However, in case of VTS, we allowed session to go beyond 25 seconds, as packets could not reach to destination within 25 seconds limit. Thus, energy consumption behavior of VTS is observed till end data transfer. Figure 5 shows that normalized energy consumption of S-MAC, T-MAC and proposed real-time MAC protocol is more or less same. It is due to fact that as wireless sensor nodes are ON all the time in S-MAC, T-MAC and proposed real-time MAC protocol, therefore, packets transmission is finished quite early in these protocols (before 4 seconds for 25 packets). Thus, all nodes were in idle listening modes for remaining session duration. However, in case of VTS, sensor nodes are ON for longer duration beyond 25 seconds, thus it has highest normalized energy consumption.

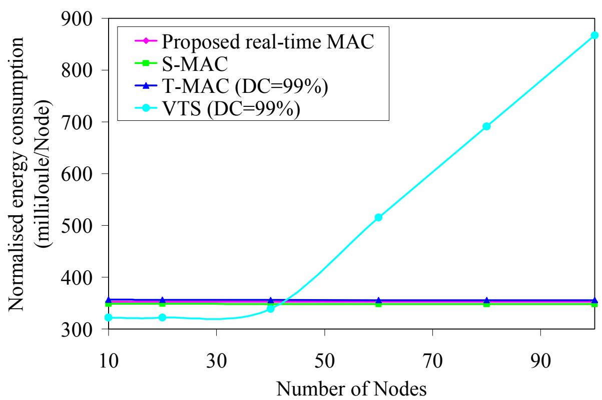


Figure 5. Normalized energy consumption pattern [Session = 25 sec. for proposed real-time MAC protocol, S-MAC, T-MAC and till end of all data transfer for VTS]

## VII. CONCLUSION AND FUTURE WORK

Real-time MAC protocol, proposed in this paper, is able to provide end-to-end delay guarantees as well as less end-to-end packet transfer delay without any significant increase in energy consumption. We found that though, packet overhead is increased in our protocol due to introduction of new control packet, however, it does not affect end-to-end packet transfer delay. Instead, our protocol have less end-to-end delay as compared to VTS, S-MAC and T-MAC protocols due to feedback based medium access strategy used in our protocol as it helps to reduced contention and collision significantly. In future, we will try to provide support for duty cycle mode of operation at MAC layer. We will also try to develop our protocol to support multi-stream wireless sensor network applications.

## REFERENCES

[1]. YE, W., HEIDEMANN, J. AND, ESTRIN, D. 2004, Medium access control with coordinated adaptive sleeping for wireless sensor networks, *IEEE/ACM Transactions on Networking, Volume 12, Issue 3, June 2004,* 493-506.

[2]. VAN DAM, T., AND LANGENDOEN, K. 2003, An Adaptive EnergyEfficient MAC Protocol for Wireless Sensor Networks, *1st international conference on Embedded networked sensor systems, SenSys'03, November 5-7, 2003*, 171-180.

[3]. LU, G., Krishnamachari, B., and Raghavendra, C.S. 2004, An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks, Proceedings of the 18th IEEE International Parallel and Distributed Processing Symposium, IPDPS 2004, 3091-3098.

[4]. LU, G., KRISHNAMACHARI, B., AND RAGHAVENDRA, C.S. 2007, An adaptive energy-efficient and low-latency MAC for tree-based data gathering in sensor networks, *Wireless Communications and Mobile Computing 7, 7,* 863-875

[5]. EGEA-LÓPEZ, E., VALES-ALONSO, J., MARTÍNEZ-SALA, A.S., GARCÍA-HARO, J., PAVÓN-MARIÑO, P., AND BUENO DELGADO, M.V. 2008, A wireless sensor networks MAC protocol for real-time applications, *Personal and Ubiquitous Computing, Volume 12, Issue 2, February 2008,* 111-122.

[6]. EGEA-LÓPEZ, E., VALES-ALONSO, J., MARTINEZ-SALA, A.S., GARCÍA-HARO, J., PAVÓN-MARIÑO, P., AND BUENO-DELGADO, M.V. 2006, A real-time MAC protocol for wireless sensor networks: Virtual TDMA for Sensors (VTS), *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 3894 LNCS,* 382-396.

[7]. CACCAMO,M., ZHANG, L. Y., SHA, L., AND BUTTAZZO, G. 2002, An implicit prioritized access protocol for wireless sensor networks, *IEEE Real-Time System Symposium RTSS 2002,* 39-48.

[8]. CACCAMO,M., AND ZHANG, L. Y. 2003, The capacity of implicit EDF in wireless sensor networks, *15th IEEE Euromicro Conference on Real-Time Systems ECRTS 2003,* 267-275.

[9]. WATTEYNE, T., AUGÉ-BLUM, I., AND UBÉDA, S. 2006, Dual-Mode Real-Time MAC protocol for Wireless Sensor Networks: a Validation/Simulation Approach, *ACM International Conference Proceeding Series 138, art. no. 1142683, Proceedings of the First International Conference on Integrated Internet Ad hoc and Sensor Networks, InterSense '06.*

[10]. WATTEYNE, T., AND AUGÉ-BLUM, I. 2005, "Proposition of a hard real-time MAC protocol for wireless sensor networks", *Proceedings of IEEE Computer Society's Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems, MASCOTS 2005, art. no. 1521180,* 533-536.

[11]. KROHN, A., BEIGL, M., DECKER, C., AND ZIMMER, T. 2005, Tomac- real-time message ordering in wireless sensor networks using the MAC layer, *In Proceedings of the 2nd International Workshop on Networked Sensing Systems, INSS 2005*.

[12]. CHEN, J., ZHU, P., AND QI, Z. 2007, PR-MAC: Path-oriented Real-time MAC protocol for wireless sensor network*, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 4523,* 530-539.

[13]. LI, H., SHENOY, P., AND RAMAMRITHAM, K. 2005, Scheduling messages with deadlines in multi-hop real-time sensor networks, *Proceedings of the IEEE Real-Time and Embedded Technology and Applications Symposium, RTAS 2005*, 415-425.

[14]. JUNGSOOK, K., JAEHAN, L., PELCZAR, C., And BYUNGTAE, J. 2008, *RRMAC: A sensor network MAC for real time and reliable packet transmission*, Proceedings of the International Symposium on Consumer Electronics, ISCE 2008,April 2008, art. no. 4559491, 1-4.