# Streaming Optimized MAC (SOMAC) for WSN

Sudheer Kumar NV
Assistant Professor
FCE, MCTE,
Mhow, MP, India - 453441

Brijendra Kumar Joshi
Professor
FCE, MCTE,
Mhow, MP, India -453441

## ABSTRACT
Wireless Sensor Networks (WSN) has seen almost a decade of active research. The concept of Wireless Multimedia Sensor Networks has been derived from WSN where the focus has shifted from low data rate random infrequent transmissions to high data rate regular transmissions in the network. In order to support multimedia transport like voice streaming in this highly constrained network, the protocols need to be optimized for the channel throughput and lifetime of the network required by the streaming application. The medium access control (MAC) protocol plays an important role in the operation of this network to avoid collisions and allocate the optimum throughput to nodes in the streaming path. A MAC protocol for a streaming voice application is proposed in this paper which allows streaming at maximum data rate in a multihop path by adapting the frame time and avoiding the collisions to the streaming nodes by the neighboring nodes. The simulation results establish the possibility of compressed voice streaming across a large distance through multihop route.

## General Terms
Algorithm, Networks, Wireless.

## Keywords
Medium Access Control, MAC, real-time, Streaming, W**SN,** Wireless Sensor Networks.

## 1. INTRODUCTION
Wireless Sensor Networks (WSNs) have been in the forefront of research in the current decade. In the initial stages of evolution, WSN comprised of low end miniaturized processing nodes with interfaced environmental sensors and low data rate radios deployed in large numbers in a random manner in the area of interest for information gathering. A node in WSN (mote) consisted of three main components: the processor, sensors and radio. The unique requirement of applications of WSN was that they had to be energy efficient within the constraints of a low memory, slow and low end processor and a low data rate radio. Therefore, generally, low data rate environmental sensing applications that generate data randomly had been attempted on such networks. The protocols in these networks were optimized to enhance the overall life time of the network relaxing other parameters like delay. With the advancements in processor and radio technologies, multimedia transport has become a possibility within the paradigm of WSN. These networks called Wireless Multimedia Sensor Networks (WMSN) have to sense high volume of data which is audio or video, process them and transport them to the destination, even if for a short duration in the lifetime of the network. An example application is sensing for voice information in an area of interest covered by the deployed sensor nodes and to transport the gathered information to a base station. This paper describes modification of an existing MAC protocol for low data rate WSN to support a streaming application where compressed voice streaming is done from an area of activity to a destination through multiple routing nodes. The protocol is specific to the specific application scenario described in section 2 and the assumptions with justifications are also brought out in section 2. The related work is surveyed in section 3. The proposed scheme is described in section 4. The simulation scenario is discussed in section 5 and the analysis of results is carried out in section 6. The paper is concluded in section 7 with the suggested future works.

## 2. ASSUMPTIONS
The SOMAC protocol makes the following assumptions regarding the application scenario and the network conditions.

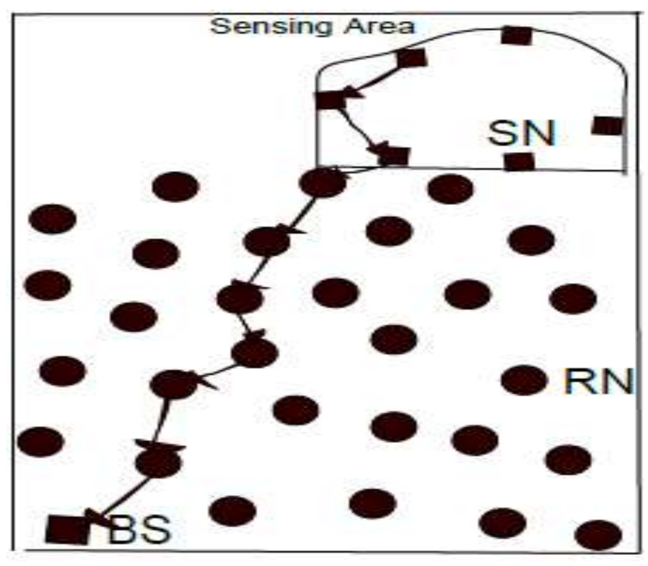


**Figure 1: Application scenario for Voice Streaming**

### 2.1 Application Scenario
In terrorism infested society, occupation of a vital infrastructure for bargain has been a common modus operandi by the militants. The law enforcement agencies then have to undertake the mission to clear the place of occupation by engaging the militants through combat. The access to information from within the area of occupation could be of immense value to this operation and WSN holds promise in such a situation to unobtrusively gather and transport data from the area of occupation to the control center of operation a few kilometers away. The area of interest

is assumed to be placed with sensor nodes (SN) fitted with acoustic (microphone) and image sensors (camera). These nodes can be pre-installed in anticipation because of the importance and vulnerability to such an event or can be fired in when the need arises. The gap between the SN to the control center where the base station (BS) is placed would be deployed with routing nodes (RN) to gather information through multihop routing. A typical application scenario is shown in figure1. The SNs would be gathering voice information and streaming will be initiated by BS by an appropriate command from one of the SNs. Any time, only one streaming path would be maintained in the network. BS would terminate the ongoing streaming session by an appropriate command.

## 2.2 Protocol Assumptions

**SO**MAC is based on the assumption of a distributed TDMA algorithm similar to the one in [1]. The protocol assumes that the synchronization of the nodes would be done with reference to the base station during the initialization phase of the network application as in LMAC [2]. SOMAC provides connectivity information to the routing protocol SPSOR [3] for establishing the streaming path from SN to BS. A multihop path is assumed to exist between SN and BS when streaming is to begin in the network. It is assumed that a node which is not a part of the streaming path could be maintained in a low duty cycle TDMA scheme whereas the nodes being part of the streaming path would be on a high duty cycle TDMA scheme to transmit at the desired voice streaming data rate. The interference does not affect the reception in nodes two hops away.

# 3. RELATED WORK

LMAC [2] is an energy efficient medium access protocol which is collision free in a time slot synchronized self organizing network. The protocol in the past has been extended for optimizing the performance parameters as in AILMAC [4] and Crankshaft [5] protocols. In LMAC, the time is divided into frames, each frame consisting of a number of time slots. The slots are maintained as per a distributed algorithm as in [1]. A node is assigned a slot without causing collision to any other node in minimum two transmission hops as per the algorithm. A slot consists of one control packet and data packet transmissions. Every node transmits control packet in its slot and every node listens to all control packets in the beginning of slot. This is to maintain the synchronization in all nodes. A node addressed in a control packet would listen to the data packet in that slot. If a collision is encountered, it is resolved by changing ownership of the slot. In dense deployments, some nodes would not be able to get collision free slots and would not be able to transmit.

In the proposed SOMAC protocol, nodes synchronize as in LMAC. Depending on the type of packet received, a node enters streaming state. In streaming state, the node discards the LMAC scheme and transmits every 3rd slot, receives in the next 2 slots following its transmission slot and does not sleep. In addition, the streaming slot time is reduced considerably from the normal slot time to accommodate higher data rate and reduced latency. The streaming slot time is maintained such that there would be a control packet transmission from the streaming nodes in the beginning of every normal slot time. The nodes not being part of the streaming state would continue to be in LMAC mode, but they would not be able to transmit the control packet in their slots if they are in the vicinity of a streaming path node. They will receive the control packet and get to know about streaming in the vicinity from the packet type field. They will wait for streaming to be over in case they have data to transmit. Hence collision in streaming mode is avoided.

In AI-LMAC [4] protocol, LMAC is adapted for a special kind of application, where different parts of a network require different data rate transmissions to a base station at different times. The protocol captures information about traffic patterns in the network and MAC operation is modified according to this data. The application envisages simultaneous access of low rate data and certain high rate data transmissions to the root node. The application adapts to these data rates. The nodes dynamically change its duty cycle depending on the amount of data flowing through the network. It is not meant for streaming, where regular flow of data from a node to BS can be ensured. AILMAC allows a node to own multiple slots; similarly, SOMAC owns multiple slots in a frame, the transmission slot being a function of the hop distance of the node to the BS. In SOMAC, the frame size is reduced effectively to accommodate only the streaming users in a frame. The proposed protocol ensures that only one streaming path is maintained, preventing transmissions in the vicinity of the streaming path nodes, allowing multiple slot operation and slot time adaptation in a frame to increase the data rate.

Crankshaft MAC protocol is specifically designed to perform well in dense deployments. The Crankshaft protocol divides time into frames, and each frame is divided into slots as in LMAC. There are two types of slots in the Crankshaft protocol: broadcast slots and unicast slots. During a broadcast slot all nodes wake up to listen for an incoming message similar to the control packet in LMAC. Any node that has a broadcast message to send contends with all other nodes to send that message. The frame starts with all the unicast slots, followed by the broadcast slots in the Crankshaft protocol. Crankshaft uses a Data/Ack sequence for unicast messages, and the slot length is such that it is long enough for the contention period, maximum-length data message and acknowledgement message. If the sender does not receive an acknowledgement, the protocol is set to retry each message three times in subsequent frames. Since streaming applications do without retransmission, ACK is not explicitly needed and Crankshaft cannot realize maximum throughput in the channel. Sink nodes will listen to all unicast slots in the Crankshaft protocol. The unicast slot concept of Crankshaft protocol is used in our protocol during streaming as it is reserved for a particular link and neighboring nodes do not transmit in these slots.

The mechanisms employed by LMAC to achieve synchronization, framing and slotting are used in Crankshaft. In our protocol, synchronization is achieved in a similar way, starting from the BS. In our protocol, a node remains in the synchronization phase for only a time out duration after switch on. After the time out period, a node will transmit only in its slot selected by a contention resolution algorithm. In SOMAC, each node keeps track of its *hop-distance* to a designated gateway node and broadcasts this information efficiently in its control message. In our protocol, every message transmitted from the node will contain the hop-distance field for node synchronization and for routing.

The PMAC [6] protocol is similar to the Crankshaft protocol in that it also schedules nodes to be awake for reception on a slot basis. However, the PMAC protocol requires nodes to exchange and store schedules. In our protocol, the contention during streaming is avoided by listening to the control packet to know the presence of streaming in the neighborhood and abstaining from transmission in the channel. The IEEE 802.15.4 MAC protocol [7] provides guaranteed time slots and contention time slots in the frame. The guaranteed time slots are decided by the beacon transmission and works well in the star topology. The reservation of time slots for streaming data transmission has been taken from this concept. However, there is no requirement of beacon transmission or explicit synchronization messages in the functioning of contention free slots in our protocol after the network set up time.

# 4. SOMAC ALGORITHM

On switch on, the nodes will synchronize to the transmissions from a BS. The mechanisms employed by LMAC to achieve synchronization, framing and slotting are used in our protocol as well. The transmission slot in a TDMA frame including control packet (CP) and data packet is shown in figure 2a. The hop distance to the BS is transmitted by every node in its control packet, listening to which, every node in the higher layer can calculate its hop distance to BS. During the setup phase, the nodes between SNs and BS would form several layers in accordance with their multihop count to the BS. When there is no streaming in the network, the nodes would maintain a longer slot time and frame time to save battery power. The routing layer finds the address of the next hop node towards BS during the setup phase. Nodes in streaming mode create subslots out of a normal slot as shown in figure 2b.

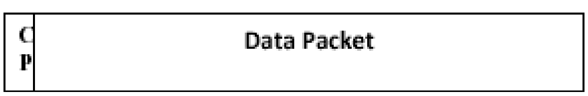


**Figure 2a :Slot Structure During Non-streaming Operation**

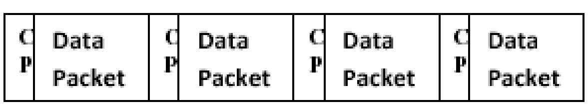


**Figure 2b :Slot Structure During Streaming Operation**

When streaming starts, the nodes will maintain a TX (Transmit Slot), RX (Receive Slot), ACK (Acknowledgement Slot) cycle for their slots and for a node, the state of the radio in a slot is a function of its layer number or distance from the BS. The frame structure of the streaming nodes is as shown in figure 3. The non-streaming slot time is divided into subslots suitable to carry the payload from the streaming source. However, the subslots are so timed that there would be a control packet transmitted by a streaming node in the beginning of every nonstreaming slot. This is to ensure that non-streaming nodes listen to the control packet in the beginning of their slot and refrain from transmission in the neighborhood of streaming nodes and also that they do not lose synchronization. There is no explicit acknowledgement transmission during streaming. However, when a node in Layer N transmits to Layer N-1, it acts as an implicit acknowledgement for Layer N+1, therefore, ACK is only a receiving slot. The relationship between Layer Number (LN) of a node and its slot type in streaming is as given by equations 4.1 to 4.3.

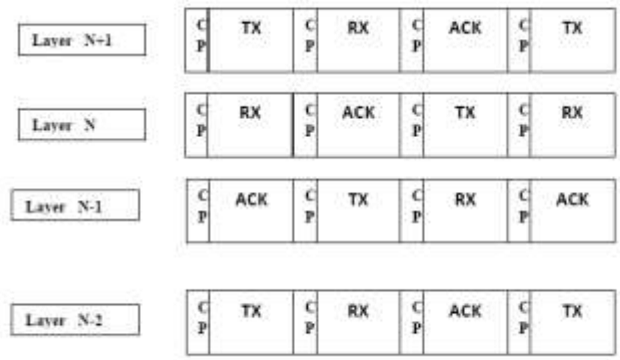


**Figure 3: Frame Structure of Streaming Nodes vs LN**

$$\text{RX Slot} = LN \% 3 \quad (4.1)$$

$$\text{ACK Slot} = (LN+1)\%3 \quad (4.2)$$

$$\text{TX Slot} = (LN+2)\%3 \quad (4.3)$$

For example, if the Layer number of the node is 13, in the TDMA frame of 3 slots numbered 0, 1, and 2, its RX slot would be 1, ACK slot would be 2 and TX slot would be 0. The number of subslots per slot should follow the formula 4.4.

$$T = 3 * (N+1) * t_s, \quad (4.4)$$

where T is the slot time, $t_s$ is the subslot time and N is an integer 0,1,2,3,……

Once the streaming is over, the frame structure returns to the original one again to save power. The nodes in the streaming path maintain a timer to indicate the time between two packets transmitted during streaming. If this time exceeds above a threshold, the node returns back to non-streaming state.

As in LMAC protocol, every node would wake up during the slot beginning and wait for control packet if that slot is not its transmission slot in the TDMA frame. If a node receives a control packet with the type field indicating streaming, the node does not transmit in its transmission slot in the TDMA frame. This avoids collisions in the streaming path. Streaming node does not have to sense the channel before transmission in its slot whereas a nonstreaming node will sense the channel before transmitting control packet in its transmission slot. This will also prevent a nonstreaming node to transmit in its slot if that node happens to be in the interference range of the streaming nodes.

## 5. SIMULATION SETUP

Mobility Framework [8] in OMNeT++4.0 [9] simulator is used for the simulation of the SOMAC protocol. 200 nodes were randomly deployed in an area of 100m x 5000m. These nodes formed 38 layers from the designated source node to BS. The nodes were designed to have a constant data rate application layer (SensorApplLayer in the Mobility Framework) with one of the nodes in layer number 37 selected to generate packets that are representative of compressed voice generated at the rate of 23.3 kbps. The radio was set to be operating on 2.45GHz at a data rate of 250 Kbps. The CC2420 radios available for WSN nodes have the above parameters. The SOMAC layer and fixed mobility model were used in the hosts of the network along with simple physical layer model of the mobility framework. In SOMAC, the nonstreaming slot time was kept at 0.18s and subslot time at 0.006s, resulting in N = 9 for equation 4.4. Every streaming node gets its transmission slot in 0.018s and hence the application data generation rate has to be at the most equal to one packet in 0.018s. The SPSOR routing layer was selected for the network layer which created a single streaming path through the layers from the source to the BS during the application set up phase. The simulation was run for 5000 seconds to generate and transmit 5000 packets at a rate of 55.55 packets per second (packet time of 0.018s). The application was scheduled to start the data generation after 2500 seconds of initialization and each packet was of 100 bytes.

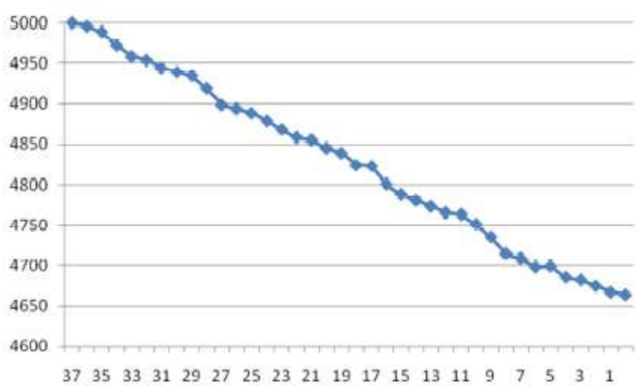


**Figure 4: Layer Number (LN) vs Number of Packets Received**

The route was set up from node in layer 37 to the BS in 37 hops by the routing algorithm during the setup phase of the application layer. The first application layer packet was sent by the designated sensing node in its streaming transmission slot as per equation 4.3. On reception of this packet, the next hop node comes to the streaming state in the next nonstreaming slot and continues to forward the received packets. The number of packets received by each of the streaming nodes in different layers is shown in figure 4. Since a node takes minimum one slot time to turn to streaming state, some of the packets transmitted from the layer above would be lost. From the node in layer 37 to BS in layer 0, the total number of packets lost is 336. It means that the application requires 336 packet times to stabilize the operation.

## 6. ANALYSIS OF RESULTS

The application layer was generating data packets of 100 bytes in every 0.018 s at a constant rate, like a compressed voice source. Considering FEC of 1//2, and the packet overheads, 40 byte payload is assumed in every 0.018s resulting in a source data rate of 23.3 kbps or less which a moderate voice codec would be able to achieve. With the network and MAC layer overheads, the packet transmission would be 120 bytes in 0.006s, resulting in a minimum data rate requirement of 160kbps from the radio, which a CC2420 radio available in practical WSN nodes. The simulation setup will be representative of a voice streaming application with compressed data rate of 23 kbps through a network of nodes with CC2420 radios.

The packets delivered to the BS show an average latency of 0.0125s per layer, which is equal to two time slot durations. With 37 layers, the latency was 0.449s which is acceptable for a one way voice link. The drop in number of packets delivered to the BS is due to the fact that streaming setup takes one slot time per layer and on an average the nodes would drop 9 packets in one slot time. The setup delay for the source to sink transmission would be number of layers times the slot time.

## 7. CONCLUSION AND FUTURE WORK

Streaming applications in WSN need optimized protocols to circumvent the constraints of hardware available. The routing and MAC layers play the most important roles in achieving the optimum throughput for the application in these networks. A MAC protocol optimized for streaming is designed over the LMAC protocol which is a TDMA based protocol with an internal time synchronization scheme.

In voice streaming applications, the nodes in the multihop streaming path have to transmit and receive packets at a constant rate depending on the compression rate. The constraints of WSN nodes demand techniques to avoid traffic around the streaming nodes. In this protocol, the nonstreaming nodes suspend the transmission on detecting an ongoing streaming activity in their hearing range. The SOMAC protocol also adapts the slot time for streaming without affecting the synchronization of any of the network nodes. The protocol has been simulated in OMNeT++4.0 simulator and the results validate the possibility of voice streaming in WSN with nodes of the iMote2 [10] variety having CC2420 radio.

The simulation model has not included an explicit battery model to account for energy constraints of the WSN nodes. Also, the propagation model used is simple and other statistical models need to be used in future simulations. Finally, the protocol has to be tested on real platforms like the iMote2 nodes, porting the protocol in NesC for TinyOS.

## 8. REFERENCES

[1] T.Nieberg, S.Dulman, P.Havinga, L.van Hoesel and J.Wu, "Collaborative Algorithms for Communication in Wireless Sensor Networks", Ambient Intelligence: Impact on Embedded Systems, Kluwer Academic Publishers, ISBN 1-4020-7668-1, November 2003.

[2] Van Hoesel, L., Havinga, P.: A lightweight medium access protocol (LMAC) for wireless sensor networks. In: Proc. of the 1st Int. Workshop on Networked Sensing Systems (INSS 2004), Tokyo, Japan (2004).

[3] Sudheer Kumar NV, BK Joshi, Single-path Streaming Optimized Routing Protocol for Dense Wireless Sensor Networks, IJANA, 2010.(Submitted).

[4] S. Chatterjea AND L. van Hoesel AND P. Havinga AI-LMAC: An Adaptive, Information-centric and Lightweight MAC Protocol for Wireless Sensor Networks, 2nd Int. Conf. on Intelligent Sensors, Sensor Networks and Information Processing, 2004.

[5] G.P. Halkes and K.G. Langendoen, Crankshaft: An Energy-Efficient MAC-Protocol for Dense Wireless Sensor Networks, European Workshop on Wireless Sensor Networks, 2007, 228-244.

[6] Zheng, T., Radhakrishnan, S., Sarangan, V.: PMAC: an adaptive energy-efficient MAC protocol for wireless sensor networks. In: Proc. of the 19th IEEE Int. Parallel and Distributed Processing Symp. (IPDPS'05). (2005) 65–72.

[7] http://standards.ieee.org/getieee802/download/802.15.4a-2007.pdf.

[8] W.Drytkiewicz, S.Sroka, V.Handziski, A.Koepke, H.Karl, *A Moibility Framework for OMNeT++*, 3rd International OMNeT++ Workshop, January 2003.

[9] A. Varga, *The OMNeT++ discrete event simulation system*, In Proc. of the European Simulation Multiconference (ESM 2001), Prague, Czech Republic, June 2001.

[10] L.Nachman, J.Huang, J.Shahabdeen, R.Adler, R.Kling, IMOTE2: Serious Computation at the Edge, IWCMC international, 1118-1123, 2008.