

Ferry Route Design with MAC Protocol in Delay Tolerant Networks

Chaitra T K¹, Dr A S Poornima_{ph.D}²

¹Student, Department of Computer Science and Engineering,

Siddaganga Institute of Technology, Tumkur - 572103, Karnataka, India.

² Associate Professor, Department of Computer Science and Engineering,

Siddaganga Institute of Technology, Tumkur - 572103, Karnataka, India.

Abstract: Delay Tolerant Networks(DTNs) are occasionally connected networks. They have high latency, long queuing time, limited resources and intermittent connectivity, which are different from traditional networks. They have been proposed to cope with challenges of communication in some extreme or special environments. Due to uncertainty of node mobility, application traffic demand and other factors, it is difficult to provide performance guarantee for a DTN where all nodes move arbitrarily. With controlled mobility, message ferry can be utilized to guarantee the network performance. MAC protocols developed for duty-cycled networks such as B-MAC,S-MAC, employ an extended preamble introduces X-MAC employs shortened preamble approach that retains the advantages of low power listening, namely low power communication, simplicity and a decoupling of transmitter and receiver schedules. Demonstrate through implementation and evaluation that xmac's shortened preamble approach significantly reduces energy usage at both transmitter and receiver, reduces per-hop latency.

Keywords: Delay Tolerant Network, Message Ferry, MAC, Shortned preamble.

I. Introduction and Motivation

Source and destination nodes do not need to be connected through a multi-hop link at a given time, a new class of mobile ad-hoc networks is emerging, Instead data packet eventually reaches its destination by passing data between nodes that come into close contact with one another. Sparse, dynamic, and end-to-end transmission delays in the data this networks are well suitable for these networks. This new class is referred to as delay tolerant networking (DTN). For DTN architecture the mobility of network nodes is an essential component. In partially-connected networks node mobility enables message passing where standard adhoc networking in which from source to destination wireless nodes cooperate to relay packets over multiple relay hops fails. When long delays can be tolerated, mobility-based data packet delivery is known to be effective. Assume nodes have similar capabilities approaches that consider given node movement typically, nodes move in random patterns, and the communication is separated from the other activity of the node. To overcome network partitioning, due to sparse numbers of nodes and reduced network congestion is the main advantages of these approaches. And their ability is data is physically carried rather than transmitted multiple times through relays . Mobility can also save overall power in the network depending on the transmission protocols. Due to their inherent flooding nature, encounters to relay data and scalability issues these is main drawbacks of these approaches.

Both communication and mobility capabilities will be combined in more devices as technology rapidly progresses. For example, a wide range of applications, mobile robots have been developed such as military, disaster recovery, home and factories. Robotic prototypes are also developing by researchers, such as robotic insects that can fly or walk on water. It is anticipated that devices with communication capabilities and mobility will become more popular in the future. In addition, mobility of the device can be achieved via the other entities movement. For example, people movements, the required mobility for the carried devices may provided by ground or aerial vehicles.

The interaction between devices effective and control of the network or the mobility of devices is needed to fully realize these benefits.

For the design of communication protocols developed for wireless sensor networks (WSNs), energy efficiency is a fundamental theme including routing and MAC layer protocols. Duty cycling is one of the primary mechanisms for achieving low energy consumption in energy-constrained WSNs. In this approach, is has two states an awake state and a sleep state. Each sensor node periodically cycles between these two states. Sleep time, wake time, and the energy consumed during the awake state and the sleep state are key parameters that characterize the duty cycle include. The sleep time plus awake time is equivalent to period of a duty cycle. Given duty cycling sensor nodes, as nodes are waking and sleeping in the network, the challenges faced by designers of communication protocols, are how to achieve

high throughput, low delay, and energy efficiency. This paper focuses on the design of X-MAC, for duty cycled WSNs an energy-efficient MAC layer protocol, and to adaptively select sleep times introduces an optimization for improved energy consumption and latency.

For duty-cycled WSNs standard MAC protocols are developed and these can be roughly categorized into synchronized and asynchronous approaches, along with hybrid combinations. These approaches are motivated by the desire to reduce idle listening, which is the time that even though no packets are being transmitted to that node, the node is awake listening to the medium. S-MAC and T-MAC are Synchronized protocols, negotiate a schedule within a frame that specifies when nodes are awake and asleep. In order to communicate reduces the time and energy wasted in idle listening by specifying the time when nodes must be awake. B-MAC and WiseMAC are asynchronous protocols, rely on low power listening (LPL), also called preamble sampling, to a receiver who is duty cycling links together a sender with data. Asynchronous protocols reduces idle listening by shifting the burden of synchronization to the sender. When a sender has data, the sender transmits a preamble that is at least. The receiver will wake up, detect the preamble, and stay awake to receive the data. This allows low power communication without the need for explicit synchronization between the nodes. The receiver only wakes for a short time to sample the medium, thereby limiting idle listening. The combination of synchronous protocol and asynchronous protocol is Hybrid protocol, like T-MAC with asynchronous low power listening is Hybrid protocol

A key advantage of asynchronous low power listening protocols is that the sender and receiver can be completely decoupled in their duty cycles. The simplicity of this design removes the need for, and the overhead introduced by, synchronized wake/sleep schedules. Studies of low power listening have demonstrated its energy-saving capabilities.

II. Related Work

In the literature there are a number of approaches to duty-cycling MAC protocols. These approaches can be broadly divided into two categories: synchronization techniques to assure that the wake periods of the nodes are concurrent; and those that have no synchronization requirements and instead depend on an extended preamble and low power listening.

S-MAC is a low power RTS-CTS based MAC protocol to allow for duty cycling in sensor networks that makes use of loose synchronization between nodes. To achieve low power duty cycling the protocol uses three techniques: periodic sleep, virtual clustering, and adaptive listening. The nodes in

the network periodically wake up, receive and transmit data, and return to sleep. To assure that the node and its neighbors wake up concurrently a node exchanges synchronization and schedule information with its neighbours at the beginning of the awake period. This schedule is only adhered to locally, resulting in a virtual cluster, which mitigates the need for system-wide synchronization. Nodes that lie on the border of two virtual clusters adhere to the schedules of both clusters, which maintains connectivity across the network. After the synchronization information is exchanged, using RTS-CTS the nodes transmit packets until the end of the awake period and the nodes then enter sleep mode. In[17], to reduce latency the authors introduce adaptive listening. From its neighbor a node hears an RTS or CTS, it will wake up briefly at the end of the transmission. On the data path, if the node is the next hop, waking up at the end of the transmission will reduce latency as the packet can be forwarded immediately without having to wait until the next scheduled awake period.

By shortening the awake period if the channel is idle introduces new protocol T-MAC improves on the design of S-MAC. In S-MAC, through the entire awake period the nodes will remain awake even if they are neither sending nor receiving data. T-MAC improves S-MAC after the synchronization phase, by listening to the channel for only a short time, and during this window if no data is received, the node returns to sleep mode. If data is received, the node remains awake until no further data is received or the awake period ends. The authors show that, T-MAC uses one fifth of the energy used by S-MAC for variable workloads. While this adaptive duty cycling reduces energy usage for variable workloads, these gains increased latency and come at the cost of reduced throughput. A comparison of duty cycling MAC protocols for WSNs is performed in [9]. Specifically, to standard CSMA/CA and LPL, S-MAC and T-MAC are compared. To use low power listening during the awake period S-MAC and T-MAC are also modified, which further decreases the energy consumption of the protocols. While they show that T-MAC in combination with low power listening provides very low power communication, latency is not considered. In addition, T-MAC as heavy a load as LPL was not able to handle and due to the early sleeping problem S-MAC.

B-MAC, developed at the University of California at Berkeley, is a CSMA-based technique to achieve low power communication that utilizes low power listening and an extended preamble. Each node can have an independent schedule and nodes have an awake and a sleep period. If a node wishes to transmit, it precedes the data packet with a preamble that is slightly longer than the sleep period of the receiver. During the awake period, a node samples the

medium and if a preamble is detected it remains awake to receive the data. With the extended preamble, a sender is assured that at some point during the preamble the receiver will wake up, detect the preamble, and remain awake in order to receive the data. B-MAC also provides an interface to changing traffic loads by which the application can adjust the sleep schedule to adapt. The method of adaptation is left to the application developer. The authors show that B-MAC surpasses existing protocols in terms of throughput, latency, and for most cases energy consumption. While B-MAC performs quite well, it suffers from the overhearing problem, and the long preamble dominates the energy usage.

WiseMAC [7], which is based on Aloha, to achieve low power communications uses preamble sampling in infrastructure sensor networks. WiseMAC the sender learns the schedules of the receiver awake periods and uses a similar technique to B-MAC, and schedules its transmission so as to reduce the length of the extended preamble. To achieve this, in the data acknowledgment frame the receiver puts the time of its next awake period. The next time the transmitter wants to send to that receiver it can begin the preamble only a short time before the receiver will awaken, taking into account possible clock skew. When sending the preamble this reduces the energy expended. In addition, for low traffic loads where the preamble is longer than the data frame WiseMAC repeats the data frame in place of the extended preamble. Receivers process this data frame and if the node is not the intended recipient it returns to sleep. If the node is the recipient, it remains awake until the end of the transmission and sends an acknowledgment. While WiseMAC solves many of the problems associated with low power communications, it does not provide a mechanism by which nodes can adapt to changing traffic patterns.

MAC Protocol and Message Ferry Design

Asynchronous duty cycling techniques are preferable for many applications to synchronized techniques in terms of throughput, energy consumption, and latency. In part, this is because due to synchronization they do not incur overhead. In addition, schedule information need not to be shared by asynchronous techniques and only stay awake long enough to sample the medium unless they are receiving or transmitting data. Hence, the awake period can be significantly shorter than that of synchronized methods. With a shorter awake period, asynchronous protocols can wake up more often while still maintaining a low duty cycle. This can also lead to reduced latency and higher throughput. However, the extended preamble begins to dominate the energy per packet as latency tolerance, and thus the receiver's sleep period, increases. In general, synchronized approaches may be more appropriate for applications with loose latency requirements. In [9], a modified version of T-

MAC is shown to conserve more energy than LPL in the absence of latency requirements. In this study, the period of T-MAC was 610 ms while the sleep time of LPL was 270 μ s; this would result in an order of magnitude difference in their latencies. In [13], the authors show that for a 10 hop network B-MAC outperforms S-MAC with respect to energy for latencies under 6 seconds. For these reasons, X-MAC builds upon the foundation provided by asynchronous duty-cycled MAC protocols. While asynchronous techniques perform quite well, there are a number of problems which, if mitigated, would allow for even more efficient communication. X-MAC is designed to address the following problems of low power listening: over-hearing, excessive preamble and incompatibility with packetizing radios.

Asynchronous low power listening (LPL) duty cycling a visual representation of is shown in the top section of Figure 1. When a node has data to send, it first transmits an extended preamble, and then sends the data packet. All other nodes maintain their own unsynchronized sleep schedules. When the receiver awakens, it samples the medium. If a preamble is detected, the receiver remains awake for the remainder of the long preamble, then determines if it is the target. After receiving the full preamble, if the receiver is not the target, it goes back to sleep.

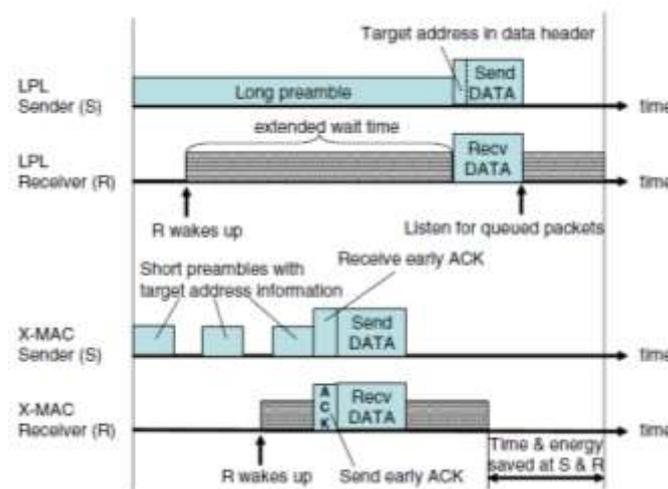


Figure Comparison of the timelines between LPL's extended preamble and X-MAC's short preamble approach.

A key limitation of LPL is that non-target receivers who wake and sample the medium while a preamble is being sent must wait until the end of the extended preamble before finding out that they are not the target and should go back to sleep. This is termed as the overhearing problem, and accounts for much of the inefficiency and wasted energy in current asynchronous techniques. This means that for every transmission, the energy expended is proportional to the number of receivers in range. Hence, the energy usage is

dependent on density as well as traffic load. This problem is exacerbated by the fact that sensor networks are often deployed with high node densities in order to provide sensing at a fine granularity. In X-MAC, we ameliorate In X-MAC, we ameliorate the overhearing problem by dividing the one long preamble into a series of short preamble packets, each containing the ID of the target node, as indicated in Figure 1. The stream of short preamble packets effectively constitutes a single long preamble. When a node wakes up and receives a short preamble packet, it looks at the target node ID that is included in the packet. If the node is not the intended recipient, the node returns to sleep immediately and continues its duty cycling as if the medium had been idle. If the node is the intended recipient, it remains awake for the subsequent data packet. As seen in the figure, a node can quickly return to sleep, thus avoiding the overhearing problem. With this technique, the energy expenditure is significantly less affected by network density. The approach of a series of short preamble packets scales well with increasing density, i.e. as the number of senders increases in a neighborhood, energy expenditure remains largely flat. In comparison, as the number of senders increase in each neighbourhood of a WSN using LPL, the entire WSN stays awake for increasing amounts of time.

Another advantage of this approach is that it can be employed on all types of radios. Any packetizing radio, such as the CC2420 characteristic of MICAz and TelosB motes, the CC2500, and/or the XBee, will be capable of sending a series of short packets containing the target ID. As we will see later, such universal support across packetizing radios is not true of the traditional extended preamble LPL. In addition, the short preamble packets can be supported across all radios with bit streaming interfaces, e.g. the CC1000 that is found in the MICA2 mote.

Low power communications is allowed using preamble sampling an extended preamble, if the total time spent transmitting preambles is reduced yet even greater energy savings are possible. In traditional asynchronous techniques, the sender sends the entire preamble even though, on average, the receiver will wake up half way through the preamble. There is no way for the sender to know that the receiver has woken up, the entire preamble needs to be sent before every data transmission. This is one case where more time is spent sending the preamble than is necessary, as illustrated by the extended wait time in Figure 1. Another case occurs for a particular receiver when there are a number of transmitters waiting to send. After the first sender begins transmitting preamble packets, until the channel is clear subsequent transmitters will stay awake and wait. They will then begin sending their preamble, and this occurs for every subsequent sender. Consequently, each sender transmits the

entire preamble when in fact the receiver was woken up by the first transmitter in the series.

In the development of X-MAC, provide solutions for both of these cases. Instead of sending a constant stream of preamble packets, insert small pauses between packets the series of short preamble packets, as would most closely approximate traditional LPL, during which time the transmitting node pauses to listen to the medium. These gaps enable the receiver to send an *early acknowledgment* packet back to the sender by transmitting the acknowledgment during the short pause between preamble packets. When a sender receives an acknowledgment from the intended receiver, it stops sending preambles and sends the data packet. This allows the receiver to cut short the excessive preamble, which reduces per-hop latency and energy spent unnecessarily waiting and a short wait time, we term this approach a strobed preamble. and transmitting, as can be seen in Figure 1. Since the sender quickly alternates between a short preamble packet In order to guarantee that preambles will be successfully received and that disconnection is avoided, the length of the preamble sequence must be greater than the maximum receiver sleep period. Additionally, the application designer may choose maximum or minimum sleep periods to bound latency and energy consumption, respectively. In addition to shortening the preamble by use of the acknowledgment, X-MAC also addresses the problem of multiple transmitters sending the entire preamble even though the receiver is already awake. In X-MAC, when a transmitter is attempting to send but detects a preamble and is waiting for a clear channel, the node listens to the channel and if it hears an acknowledgment frame from the node that it wishes to send to, the transmitter will back-off a random amount and then send its data without a preamble. The randomized back-off is necessary because there may be more than one transmitter waiting to send, and the random back-off will mitigate collisions between multiple transmitters. Also, the back-off is long enough to allow the initial transmitter to complete its data transmission. To enable this technique, after the receiver receives a data packet it will remain awake for a short period of time in case there are additional transmitters waiting to send. The period that a receiver remains awake after receiving a data packet is equal to the maximum duration of the senders back-off period, to assure that the receiver remains awake long enough to receive any additional transmitters data packet. Together, these two techniques greatly reduce excessive preambles, result in the reduction of wasted energy, and allow for lower latency and higher throughput. In addition, both of these techniques are broadly applicable across all forms of digital radios, including packetized and bit stream, because the short time

gaps, early acknowledgments, and random back-off can all be implemented in software.

In prior work, the Message Ferrying (MF) scheme for delay-tolerant networks has been proposed where end-to-end paths do not exist between some or all nodes. To overcome network partitions, to transport data the MF scheme exploits controlled mobility. Specifically, a set of special mobile nodes called message ferries move around the deployment area and are responsible for carrying data between nodes. The MF scheme can be used in different delay-tolerant environments, For example, in a disaster scene where existing infrastructure is unusable, airplanes or vehicles can be used as ferries to transport data between users in separated areas. In sensor networks where power supplies are severely limited, mobile entities such as robots or manned vehicles can be deployed to approach and collect data from sensors in order to conserve sensor energy. In MF networks, data is transported via ferry mobility. Therefore, the design of ferry movement, or ferry routes will have significant impact on network performance.

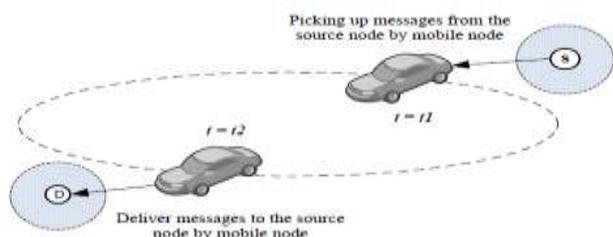


Figure 2 Message ferry node collecting data from source to destination

Implementation

Implemented the protocol on NS2 simulator in order to evaluate and demonstrate the correctness and benefits of X-MAC. NS2 is an open source, network simulator, object oriented simulator written in c++ with an OTcl interpreter as a frontend developed at UC Berkely. By calling an initialization function an application starts X-MAC which takes the minimum sleep time, maximum sleep time, initial sleep time, and initial wake time as parameters. A second initialization function, spawns a receive thread that wakes and sleeps the radio and handles the adaptation of the sleep periods. The current and boundary values for the wake and sleep times can also be modified directly.

To compare X-MAC protocol, have implemented a basic asynchronous S MAC protocol. This protocol is the closest approximation that could develop using a packetizing radio. When sending, a stream of preamble packets, transmitter sends as rapidly as possible, and after the extended preamble the data packet is sent. There are two differences between

X-MAC and the S-MAC protocol; first, for the target ID the protocol does not inspect the preamble packets so all receivers will remain awake until they receive the data packet; second, with the S-MAC protocol, receivers do not send an acknowledgement packet to the transmitter and transmitters always send the entire extended preamble and. In addition, the adaptation algorithm cannot be applied to the protocol. Although the receiver can adjust its sleep period, the transmitter will not be aware of this change so it will not know to adjust the length of its preamble.

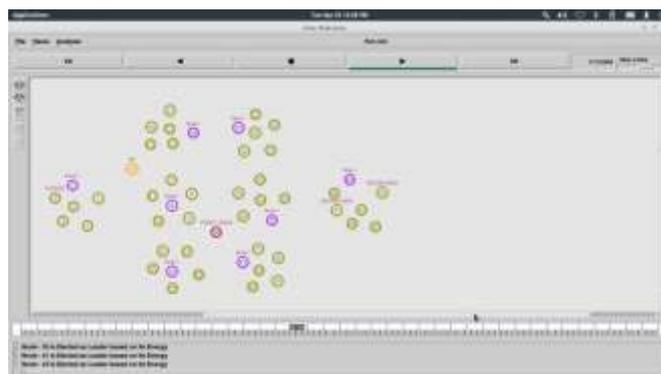


Figure 3 Message ferry node carrying data from source to destination with xmac protocol.

III. Performance Evaluation

Performed a number of experiments that test X-MAC to evaluate the performance impact of the X-MAC protocol, without adaptation on simple topologies with no contention. To demonstrate the benefits of the overhearing avoidance and the strobed preamble in X-MAC, performed an experiment with a varying number of nodes. For this, set up a cluster of nodes consisting of one receiver, one sender and message ferry node where all nodes are within transmission range of each other. Each node sends a packet once every 9 seconds to the receiving node and all nodes have a sleep period and preamble length of 500 ms. The number of senders was varied between 1 and 9, with the transmissions timed so as to avoid contention.

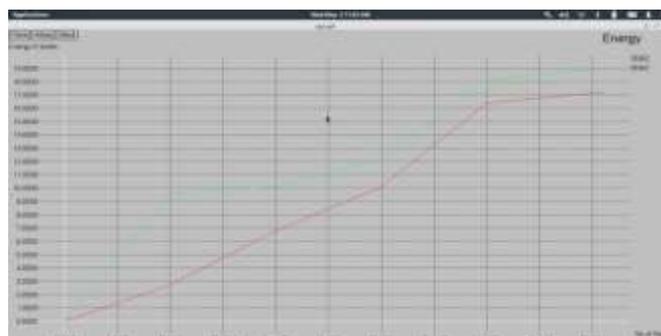


Figure 4 Comparison of energy

IV. Conclusion and Future work

A new approach to low power communication in WSNs X-MAC has been described in this paper. X-MAC employs transmitting a series of short preamble packets, a strobed preamble approach by each containing the address of the target receiver. Message ferry node carrying the data from source to destination. The series of short preamble packets approximates a continuous preamble. Small pauses between preamble packets permit the target receiver. Allows for lower latency by truncating the preamble saves energy at both the transmitter and receiver. Non-target receivers rather than remaining awake for the full preamble as in conventional LPL can go back to sleep immediately, which overhear the strobed preamble. This strobed preamble approach can be readily adapted to the packetized radios that are emerging as the standard in today's sensor motes. In future enhancement to improve the packet delivery ratio and the end to end delay by minimizing the sleep awake schedule.

References

- [1] X-MAC, Proceedings of the 4th international Conference Embedded networked sensor systems.
- [2] A multi hop reservation method for end-to-end latency performance in asynchronous MAC-based wireless sensor networks. Hong,Sung-hwa and Hoon-ki Kim
- [3] Adaptive time division multiple access based Medium access control protocol for energy Conserving. Hsu, T-H and P-Y Yen.
- [4] Performance energy efficient MAC protocols for wireless sensor networks. Roy, Alak and Nitya.
- [5] X-MAC: A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks.