MH-TRACE: MULTI-HOP TIME RESERVATION
USING ADAPTIVE CONTROL FOR ENERGY EFFICIENCY

Bulent Tavli and Wendi B. Heinzelman

Department of Electrical and Computer Engineering,
University of Rochester, Rochester, NY 14627

Abstract- Multi-Hop Time Reservation Using Adaptive Control for Energy Efficiency (MH-TRACE) is a distributed MAC protocol for energy efficient real-time packet broadcasting in a multi-hop radio network. In MH-TRACE, the network is dynamically partitioned into clusters without using any global information except global clock synchronization. The clustering algorithm is simple and robust enough to ensure that the gain from clustering is much higher than the clustering overhead, even in the presence of node mobility. In MH-TRACE, time is organized into superframes, which consist of several time frames. Each cluster chooses a frame for transmitting control packets and for the transmission of data from nodes in the cluster. However, each node in the network can receive all the desired packets in its receive range without any restriction based on the formed clusters. Each node learns about future data transmissions in its receive range from information summarization (IS) packets sent prior to data transmission by each transmitting node. Therefore, each node creates its own listening cluster and receives the packets it wants. By avoiding energy dissipation for receiving unwanted data packets or for waiting in idle mode, MH-TRACE guarantees the network to be highly energy efficient. Furthermore, since data transmission is contention free, the throughput of MH-TRACE is better than the throughput of CSMA type protocols under high traffic loads.

INTRODUCTION

Energy efficient broadcasting of streaming data, such as voice, with stringent quality of service (QoS) requirements in a mobile wireless ad hoc network is a challenging task. Although many protocols are proposed in the literature [1][2][3][4][5], neither energy efficiency nor support for real-time streaming media are completely solved issues in ad hoc networks due to the highly dynamic topologies and limited network resources. It is argued in [6] that it is hard to achieve design goals such as energy efficiency and application-specific QoS requirements by using a system consisting of independently designed layers of the protocol stack. Alternatively, a cross layer design that takes into account the specific QoS requirements of the application and tailors the rest of the protocol stack accordingly can achieve the design goals with much higher efficiency when compared to a general architecture [7].

Energy efficiency of a wireless network can be achieved by jointly optimizing the transmit power [8], minimizing idle listening periods [5], avoiding reception of collided packets [9], and avoiding overhearing irrelevant transmissions [10]. In addition, power saving mechanisms should not prevent the nodes from receiving or transmitting necessary data or control packets.

Several MAC protocols have been developed with the goal of minimizing energy dissipation of the nodes. Sensor MAC (S-MAC) [5], which is built on top of IEEE 802.11, reduces Idle listening by periodically shutting the radios off. Overhearing is avoided by entering the sleep mode after receiving the RTS and/or CTS packet until the NAV timer expires. Due to the fixed sleep time/awake time ratio, some portion of the bandwidth is always unusable and packet latency increases using this approach.

Power aware multi-access protocol with signaling for ad hoc networks (PAMAS) [10] is an energy efficient MAC protocol that uses two separate and independent channels that are capable of transmitting and receiving without creating interference for each other, for signaling and data transmissions. PAMAS avoids energy dissipation for overhearing packets destined for other nodes by shutting down the radios that are not participating in packet transmission and/or reception. However, energy dissipation in idle mode is not addressed in PAMAS.

Energy efficiency in a multi-hop network necessitates coordination between the nodes, so that they avoid idle listening or overhearing irrelevant packets or collisions, especially in broadcasting scenarios. While this goal can be accomplished using centralized control, this is not practical in a mobile ad hoc network, or at least not scalable due to the high overhead to monitor and convey the control information throughout the network. Thus, broadcast tree construction for throughput, delay, and energy optimization is not realistic and should be taken as a theoretical benchmark [11][12]. Network partitioning through clustering introduces a realizable, yet useful framework for network coordination, which has been investigated thoroughly [3][13]. We use this approach in the design of MH-TRACE.

Support for streaming media necessitates timely delivery of packets to avoid packet dropping. Voice packets can be modeled as continuous bit rate (CBR) traffic with packet delay limits. For acceptable quality, the packet drop ratio should be low. Periodic time frame based channel

allocation with automatic renewal of channel access, where frame rate is matched to the periodic rate of voice sources, lets voice flows be uninterrupted and is a natural solution for streaming media.

Several of the above techniques are utilized in the design of MH-TRACE. In MH-TRACE, the network is partitioned into overlapping clusters through a distributed algorithm, which needs very little control overhead. Time is organized into cyclic constant duration superframes consisting of several frames. Each clusterhead chooses the least noisy frame to operate within and dynamically changes its frame according to the interference level of the dynamic network. Nodes gain channel access through a dynamically updated and monitored transmission schedule created by the clusterheads, which eliminates packet collisions within the cluster. Collisions with the members of other clusters are also minimized by the clusterhead's selection of the minimal interference frame. Ordinary nodes are not static members of clusters, but they choose the cluster they want to join based on the spatial and temporal characteristics of the traffic, taking into account the proximity of the clusterheads and the availability of the data slots within the corresponding cluster. Each node that is scheduled to transmit data sends a short information summarization (IS) packet prior to actual data transmission, through which the neighbor nodes decide whether to stay awake and receive the data packets or enter the sleep mode for the duration of the data packet and avoid reception of an irrelevant or collided data packet (*i.e.*, if the IS packet is collided than it is obvious that the data packet will also collide).

When compared to CSMA-type broadcast protocols like IEEE 802.11, MH-TRACE has three advantages: (i) energy efficiency due to the use of TDMA and IS slots, which allow nodes to enter sleep mode often, (ii) higher throughput due to the coordinated channel access, and (iii) support for real-time operation due to its time-frame based cyclic operation.

MH-TRACE

MH-TRACE Operation

MH-TRACE is organized around superframes with duration, T_{SF}, matched to the periodic rate of voice packets, where each superframe consists of N_F frames. The frame format is presented in Figure 1. Each frame consists of two sub-frames: a control sub-frame and a data sub-frame. The control sub-frame consists of a beacon slot, a clusterhead announcement (CA) slot, a contention slot, a header slot, and an information summarization (IS) slot.

At the beginning of each occupied frame, the clusterhead transmits a beacon message. This is used to announce the existence and continuation of the cluster to the cluster members and the other nodes in the transmit

range of the clusterhead. By listening to the beacon and CA packets, all the nodes in the carrier sense range of this clusterhead update their interference level table.

The contention slot, which immediately follows the CA slot, consists of N_c sub-slots. Upon hearing the beacon, each node that has data to send but did not reserve a data slot in the previous cyclic superframe, randomly chooses a sub-slot to transmit its request. If the contention is successful (*i.e.*, no collisions), the clusterhead grants a data slot to the contending node. Following the contention subslot, the clusterhead sends the header, which includes the data transmission schedule of the current frame. The transmission schedule is a list of nodes that have been granted data slots in the current frame, along with their data slot numbers. A contending node that does not hear its ID in the schedule understands that its contention was unsuccessful (*i.e.*, a collision occurred or all the data slots are already in use) and contends again in the following superframe. If the waiting time for a voice packet during contention for channel access exceeds the threshold, T_{drop}, it is dropped.
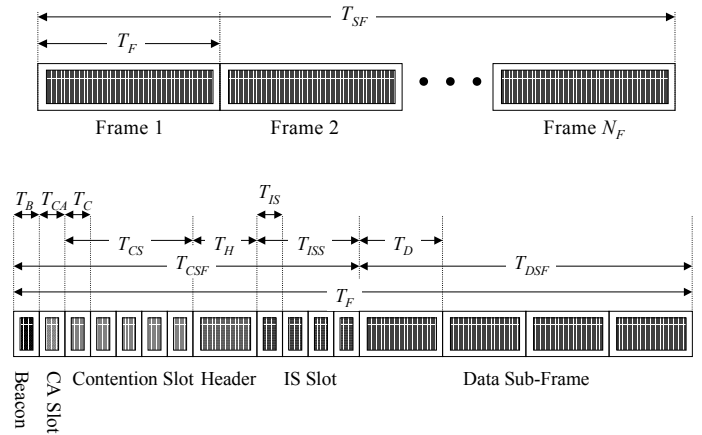

Figure 1. MH-TRACE superframe and frame formats.

The IS slot begins just after the header slot and consists of N_D sub-slots. Nodes that are scheduled to transmit in the data subframe transmit a short IS message exactly in the same order as specified by the data transmission schedule. An IS message has an end-of-stream bit, which is set to one if the node has no data to send. Each receiving node records the received power level of the transmitting node and inserts this information into its IS table. The IS table is used as a proximity metric for the nodes. Nodes that are not members of this cluster also listen to the IS slot and record the received power level. Each node creates its own listening cluster by selecting the top N_{max} transmissions that are the closest transmitters to the node. Note that other methods of deciding which nodes to listen to can be used within the MH-TRACE framework by changing what data nodes send in the IS slot. Hence the network is softly

partitioned into many virtual clusters (called listening clusters) based on the receivers.

The data subframe is broken into constant length data slots. Nodes listed in the schedule in the header transmit their data packets at their reserved data slots. Each node listens to at most N_{max} data transmissions in a single superframe; therefore, each node is on for at most N_{max} data slots.

A node keeps a data slot once it is scheduled for transmission as long as it has data to send. A node that sets its end-of-stream bit (in the IS packet) to one because it has no more data to send will not be granted channel access in the next superframe.

Cluster Formation and Maintenance

At the initial startup stage, a node listens to the medium to create its interference table. If there are clusterheads in its receive range, the node chooses the closest clusterhead with available data slots and starts its normal operation. If no beacon is detected, then the node chooses the least noisy frame and picks a random time to transmit its own beacon signal, and begins to listen to the channel until its contention timer expires. If a beacon is heard in this period, then the node just stops its timer and starts normal operation. Otherwise, when the timer expires, the node sends the beacon and assumes the clusterhead position. In case there is a beacon collision, none of the colliding nodes will know it, but the other nodes hear the collision, so the initial startup continues. All the previously collided nodes, and the nodes that could not detect the collision(s) because of capture, will learn of the collisions with the first successful header transmission. Cluster creation is presented as a flow chart in Figure 2.

Each clusterhead continuously records the interference level of each frame by listening to the beacon transmissions A clusterhead can record the interference level of each frame by listening to the beacon slot, but the beacon slot becomes useless for a clusterhead's own frame, because it is transmitting its own beacon. A CA packet, which is transmitted with a probability p_{CA}, is used to determine the interference level of the co-frame clusters.

A clusterhead keeps its frame unless another clusterhead enters in its receive range. A cluster leaves a frame with high interference and moves to a low interference frame with probability p_{CF}. The reason for adding such randomness is to avoid the simultaneous and unstable frame switching of co-frame clusters, which are the interference sources for each other. When two clusterheads enter in each other's receive range, the one who receives the other's beacon first resigns.

If a node does not receive a beacon packet from its clusterhead for $2T_{SF}$ time, either because of mobility of the node or the clusterhead or the failure of the clusterhead, then it enters the initial startup procedure.

The cluster creation and maintenance algorithm of MH-TRACE has three distinct features from existing clustering approaches [3][13]. First, MH-TRACE needs less information than the other clustering approaches, so the overhead is lower. For the clustering techniques proposed in the literature, each node needs one or more hop connectivity information to execute the cluster creation and maintenance algorithm. Almost all of these algorithms create a unique clustering for a given node distribution, thus they are deterministic. In MH-TRACE, the only information a node needs to know is the interference level, which is monitored continuously by listening to beacon and/or CA slots. However, for a given node distribution there are many clustering possibilities in MH-TRACE, thus it is probabilistic. Second, by using the interference level as a constraint for cluster creation, secondary affects are also incorporated into cluster creation, which is crucial in avoiding collisions. Interference is not considered as a constraint in the other clustering approaches. Third, there are hard boundaries between the clusters if the clusters use different spreading codes (*i.e.*, two IEEE 802.11 networks with different barker codes) or frequencies (*i.e.*, cell phones) to avoid inter-cluster interference. In these networks, even if the nodes are very close they cannot communicate directly if they are not in the same cluster. In MH-TRACE, clusters use the same spreading code or frequency, and inter-cluster interference is avoided by using time division between the clusters, which does not create any hard boundaries in the network.
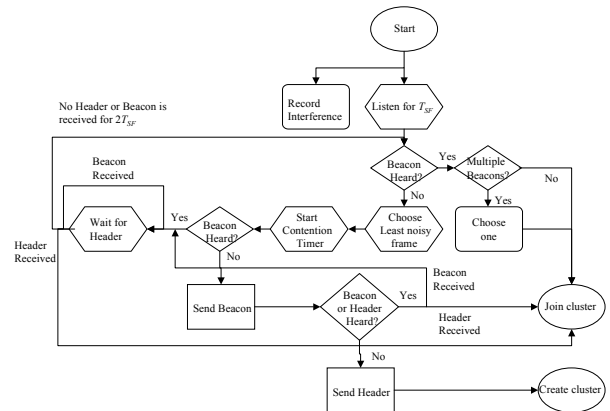

Figure 2. Cluster creation flow chart.

Energy Savings Techniques

There are two techniques used in MH-TRACE to save energy. The first technique is to reduce energy dissipation at the MAC layer. Nodes should be in sleep mode whenever possible to avoid dissipating energy in the idle state and to avoid overhearing transmissions initiated from nodes that are further than the successful transmission range. Any node in the startup mode cannot enter the sleep mode until it reaches the steady state mode. Similarly, all nodes are required to be awake for all

Beacon, CA and IS slots for all the frames within the superframe to gather the control information to run MH-TRACE seamlessly. Ordinary nodes also stay awake to receive the header slot of their own clusterhead. In addition, clusterheads stay awake in their own frames through the contention slot to receive any contention requests.

The second technique is to reduce energy dissipation using an application dependent cross layer approach, namely, avoiding packet receptions that will be discarded at the higher layers of the protocol stack if not avoided at the MAC layer. Based on the information sent in the IS slots, the MAC layer can decide whether or not to receive the data packet. If there is no discrimination of packets and all packets are to be received, then each node stays awake for all the data transmissions in its receive range, and goes to sleep mode in the data slots that are known to be empty or result in collisions through listening to the IS slots. Thus, traffic adaptive energy efficiency is achieved even without data discrimination. However, by employing data discrimination through virtual cluster creation, further energy savings can be achieved. In the simulations we used proximity, which is obtained from the receive power of the IS packets, as our discrimination metric and set a maximum size, N_{max}, on the number of virtual cluster members.

SIMULATIONS

To test the performance of MH-TRACE, we ran simulations using ns-2. We simulated conversational voice coded at 32 Kbps, which corresponds to one voice packet per superframe. The channel rate is set to 2 Mbps. Acronyms, descriptions and values of the parameters used in the simulations are presented in Table I.

Beacon, CA, contention, and IS packets are all 4 bytes. The header packet has a variable length of 4-18 bytes, consisting of 4 bytes of packet header and 2 bytes of data for each node to be scheduled. Data packets are 104 bytes long, consisting of 4 bytes of packet header and 100 bytes of data. Each slot or sub-slot includes 16 µsec of guard band (*IFS*) to account for switching and round-trip time.

For voice source modeling, we assume each node has a voice activity detector, which classifies speech into "spurts" and "gaps" (*i.e.,* gaps are the silent moments during a conversation). During gaps, no data packets are generated, and during spurts, data packets are generated in the rate of the speech coder, which is 32 Kbps in our case. Both spurts and gaps are exponentially distributed statistically independent random variables, with means m_s and m_g, respectively. In our simulations and analysis we used experimentally verified values of m_s and m_g, which are 1.0 s and 1.35 s, respectively [14].

We used the energy and propagation models discussed in [7]. Transmit, receive, idle, and sleep power dissipation are given in Table I. In the simulations we used a constant transmit power, which results in a constant transmit radius, D_{Tr}, of 250 m.

We used the random way-point mobility model to create mobility scenarios within a 1km by 1km area. Node speeds are chosen from a uniform random distribution between 0.0 m/s and 5.0 m/s (average pace of a marathon runner). All the simulations are run for 100 s and averaged five times.

Table I. Parameters used in the simulations.

Acronym	Description	Value
T_B	Beacon slot duration	32 µs
T_{CA}	CA slot duration	32 µs
T_C	Contention sub-slot duration	32 µs
T_H	Header slot duration (max)	92 µs
T_{IS}	IS sub-slot duration	32 µs
T_D	Data slot duration	432 µs
IFS	Inter-frame space	16 µs
T_{drop}	Packet drop threshold	50 ms
N_{max}	Virtual cluster size (max)	5
P_T	Transmit power	0.6 W
P_{TE}	Transmit electronics power	0.318 W
P_{PA}	Power amplifier power	0.282 W
P_R	Receive power	0.3 W
P_I	Idle power	0.1 W
P_S	Sleep power	0.0 W

Optimizing MH-TRACE Parameters

Figure 3(a) shows the average aggregate number of voice packets received per superframe, which is the total network throughput, versus the number of frames per superframe, N_F, in a 100-node network. Superframe time is adjusted to be approximately the same, 25 ms, for all four configurations (see Table II). The number of received packets is lowest, 750 ± 22, at $N_F = 4$, it reaches the maximum, 812 ± 23, at $N_F=7$, and again drops to 794 ± 12 at $N_F=8$. Figure 3(b) shows the total packet loss per superframe as a function of N_F. The packet loss (f_{pl}), which is the sum of collisions (f_{cl}) and throughput loss due to packet drops (f_{pd}), both of which are functions of N_F, explains the throughput differences in Figure 3(a). For low N_F the dominant packet loss mechanism is collisions due to low co-frame clusterhead separation. On the other hand, for high N_F throughput loss due to packet drops becomes dominant. Since the average cluster coverage area is the same for all N_F, available data slots per unit area decreases with increasing N_F. It is evident that there is a tradeoff in choosing N_F, which is created by the two loss mechanisms. The optimal value of N_F, which minimizes packet loss and maximizes throughput, is seven. Although these simulation

results are for a specific node density (*i.e.*, 100 nodes / 1 km^2), simulations with different node densities, which are not shown, also verify that the optimal N_F value is seven. We will use this value for N_F for the rest of the simulations.

Table II. Superframe parameters.

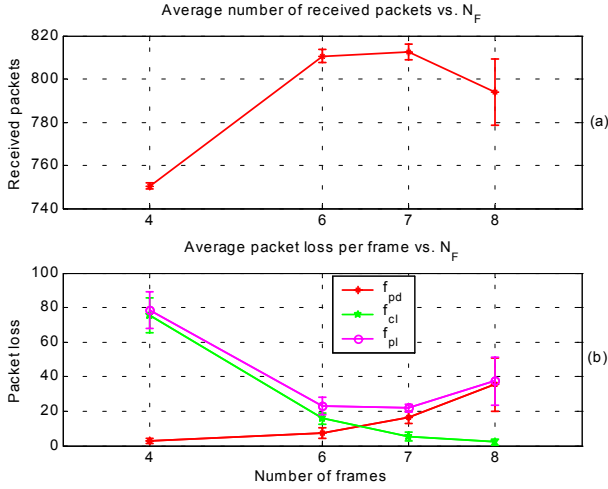| Number of frames per superframe, N_F	Number of data slots, N_D	Number of contention slots, N_C	T_F (ms)
4	12	15	24.976
6	8	9	24.984
7	7	6	25.172
8	6	6	24.992

Figure 3. (a) Average aggregate number of received packets per superframe vs. N$_F$.. (b) Average packet loss per superframe vs. N$_F$.

Throughput Performance

Figure 4(a) shows the average number of packet receptions per superframe versus the number of nodes (N_n) for MH-TRACE and 802.11. The number of received packets is very close for both protocols for $N_n = 50$, because at this node density there is not much contention for channel access and CSMA/CA is doing a good job in avoiding collisions. However, for higher N_n, 802.11's throughput starts to drop below the throughput achieved by MH-TRACE. For $N_n = 200$, MH-TRACE throughput is 2656 packets per superframe and 802.11 throughput is 1418 packets per superframe time, which is 47% lower than that of MH-TRACE. We reduced the overhead for IEEE 802.11 broadcast data packets to four bytes in our simulations to compare MH-TRACE and IEEE 802.11 on a fair basis.

Figure 4(b) shows the number of collided packets per superframe versus N_n for MH-TRACE and 802.11. In accordance with Figure 4(a), the number of collided

packets is very low for both protocols at $N_n = 50$, due to the low level of contention. For larger N_n, the number of collisions increases for both methods, but the increase in the 802.11 curve is much higher than that of MH-TRACE. At $N_n = 200$, the number of collided packets per superframe time for MH-TRACE and 802.11 are 86 and 1722, respectively.

802.11 has a constant backoff window for broadcasting, unlike the binary exponential backoff mechanism that adaptively adjusts the backoff window in unicast traffic by getting feedback from the RTS/CTS/DATA/ACK handshaking, which is not available for broadcasting due to the lack of any handshake packets. Since there is no adaptive adjustment mechanism available for broadcasting, the backoff window is chosen to be an optimal value for a particular packet size and data traffic, which maximizes channel utilization. Therefore, 802.11 cannot keep up with the varying data traffic. For example, for $N_n = 50$, the throughput obtained with 802.11 is as good as that of MH-TRACE and the delay is much lower, but for $N_n = 200$, 802.11 throughput is equal to half of the throughput obtained with MH-TRACE and the delay is still lower. For data packets, lower delay is better, but for voice packets this is not always true. A voice packet with a 50 ms delay, the maximum packet delay allowed by the MAC layer after which the packets are dropped, and another voice packet with a 1.0 ms delay are equivalent from the application's point of view, which shows that QoS is an application dependent concept and should be considered in the design of all layers of the protocol stack. MH-TRACE exploits this feature of voice packets to tradeoff the packet delay for throughput.
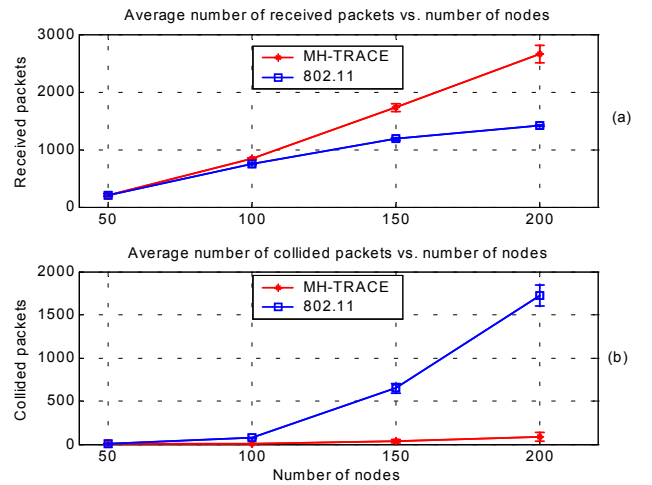

Figure 4. (a) Average number of packets received per superframe time vs. N_n. (b) Average number of collided packets per superframe time vs. N_n.

Energy Dissipation

Figure 5 shows the total network energy dissipation per superframe as a function of node density for 802.11, MH-TRACE without utilizing listening clusters, and MH-TRACE with a listening cluster size of 5 (lc-5).

Network energy dissipation for all three curves increases with the number of nodes in the network, N_n. Energy dissipation of 802.11 is due to transmissions, receptions, and idle listening. Since nodes in 802.11 listen to all the transmissions in the medium, regardless of whether the packets collide or are below the receive threshold but above carries sense threshold, its energy dissipation is much higher than MH-TRACE even without data discrimination via listening cluster creation. 802.11 energy dissipation is 334% higher than that of MH-TRACE at $N_n = 50$, because 802.11 dissipates energy for idle listening and MH-TRACE does not. At $N_n = 200$, 802.11 energy dissipation is 152% higher than that of MH-TRACE. Although energy dissipation of 802.11 is higher than MH-TRACE, the total number of received packets is lower because of the high number of collisions. 802.11 wastes energy for receiving collided packets or for carrier sense, where MH-TRACE avoids both of these situations as well as idle listening.

MH-TRACE lc-5 dissipates almost the same energy as MH-TRACE at $N_n = 50$, because the average number of transmitting neighbors is not higher than five at this node density. However, with the increasing node density, energy savings by utilizing listening clusters becomes more evident. For example, at $N_n = 200$, the energy dissipation of regular MH-TRACE is 75 % higher than that of MH-TRACE lc-5. This is because with node densities higher than 50 nodes / km^2, the number of simultaneously transmitting nodes exceeds five, which is the maximum listening cluster size.
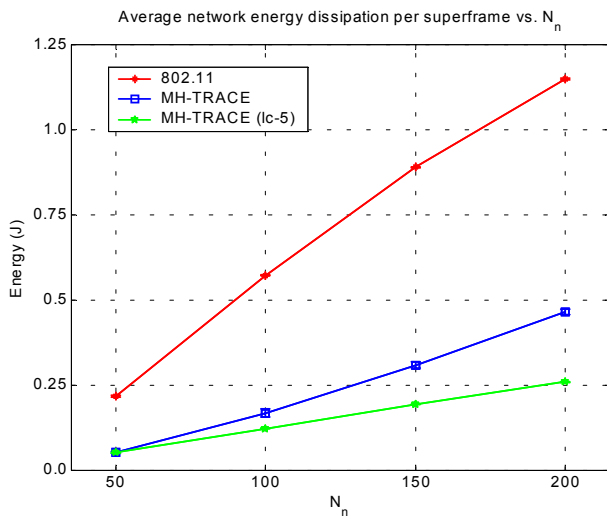

Figure 5. Average network energy dissipation per superframe time vs. N_n.

CONCLUSION

The most important advantage of MH-TRACE is that it achieves traffic adaptive energy efficiency in a multi-hop network without using any global information except synchronization. In addition, data discrimination via receiver based listening clustering creates an option for the application to save energy more aggressively. We used the cluster concept in such a way that each node creates its own listening cluster as if it is operating under a CSMA type protocol. However, collisions of data packets are also avoided by means of coordination via scheduling. Thus, advantageous features of fully centralized and fully distributed networks are combined to create a hybrid and better protocol for real-time energy efficient broadcasting in a multi-hop network.

REFERENCES

[1] A. Ephremides and T. Truong, "Scheduling broadcasts in multihop radio networks," *IEEE Trans. Commun.*, vol. 38, pp. 456-460, 1990.

[2] C. Zhu and M. S. Corson, "A five phase reservation protocol (FPRP) for mobile ad hoc networks," *Wireless Networks*, vol. 7, pp. 371-384, 2001.

[3] C. R. Lin and M. Gerla, "Adaptive clustering for mobile wireless networks," *IEEE J. Select. Areas Commun.,* vol. 15, pp. 1265-1275, Sep. 1997.

[4] S.Singh and C. S. Raghavendra, "PAMAS: Power aware multi-access protocol with signaling for ad hoc networks," *ACM Computer Communication Review*, vol. 28, pp. 5-26, Jun. 1998.

[5] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," *in Proceedings of IEEE INFOCOM'02*, pp. 1567-1576, 2002.

[6] W. B. Heinzelman, "Application-specific protocol architectures for wireless networks," Ph.D. dissertation, MIT, Cambridge, 2000.

[7] W. B. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An application specific protocol architecture for wireless microsensor networks," *IEEE Trans. Wireless Comm.*, vol. 1, pp. 660-670, 2002.

[8] V. Rodoplu and T. Meng, "Minimum energy mobile wireless networks," *IEEE J. Select. Areas Commun.*, vol. 17, pp. 1333-1344, 1999.

[9] B. Tavli and W. B. Heinzelman, "TRACE: Time Reservation Using Adaptive Control For Energy Efficiency," *to appear in IEEE J. Select. Areas Commun.*, 2003.

[10] S.Singh and C. S. Raghavendra, "PAMAS: Power aware multi-access protocol with signaling for ad hoc networks," *ACM Computer Communication Review*, vol. 28, pp. 5-26, 1998.

[11] M. Cagalj, J. P. Hubaux, and C. Enz, "Minimum energy broadcast in all wireless networks NP completeness and distribution issues," *in Proc. MOBICOM'02*, pp. 172-182, 2002.

[12] A. F. F. Clementi, P. Crescenzi, P. Peuna, G. Rossi, and P. Vocca, "On the complexity of computing minimum energy consumption broadcast subgraphs," in Proc. 18th Annual Symp. Theoretical Aspects of Comp. Science, vol. 2010 of LNCS, pp. 121-131, 2001.

[13] C. R. Dow, J. H. Lin, A. F. Hwang, and Y. W. Wang, "An efficient distributed clustering scheme for ad-hoc wireless networks," *IEICE Trans. Commun.*, vol. E85-B, pp. 1561-1571, 2002.

[14] D. J. Goodman and S. W. Wei, "Efficiency of packet reservation multiple access," *IEEE Trans. Vehic. Technol.*, vol 40, pp. 170-176, Feb. 1991.