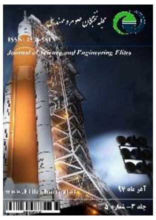
www.ElitesJournal.ir

مجله نخبگان علوم و مهندسی

Journal of Science and Engineering Elites

جلد ۳- شماره ۵- سال ۱۳۹۷



# Network Coding Aided Congestion Control in Wireless Mesh Networks

**Nasrin Safarnia[1] , Ramin Sayadi [2]**

**1-Department of Electrical and Computer Engineering, Islamic Azad University, Ilam, Iran**

**2- Department of Electrical and Computer Engineering, Islamic Azad University, Ilam, Iran**

* ramin.sayyadii@gmail.com

Received: November 2018 Accepted: December 2018

## Abstract

Routing in wireless sensor networks is one of the main challenges which directly influences the reliability and efficiency of the network. In this research, we combined several flows using inter-coding technique and hierarchical tree alternative path (HTAP) technique in order to efficiently control the congestion. It uses random network coding idea to increases network efficiency and decreases the congestion implicitly. Also, it explicitly reroutes the congested flows along alternative paths when the congestion condition is detected in congested nodes. The proposed method was simulated in different scenarios using MATLAB software and the simulation results showed that it decreases the number of transmitted packets, improves efficiency, and decreases energy consumption in comparison to original HTAP.

**Keywords: congestion control; wireless sensor networks; alternative route; network coding**

## 1- Introduction

Wireless sensor networks are composed of tiny nodes with the ability to sense a phenomenon, convert analogue data to digital data and transfer them to sink nodes. The wireless sensor nodes between the data source, which sensed the phenomenon, and the sink should form a routing path to transfer data through the path's components in a step-by-step manner. Typically, sensor nodes are placed randomly in the field in a redundant manner to cover unexpected path disconnections. Also, each sensor node is powered by a small battery, which implies that one should carefully account for power consumption considerations when designing a routing mechanism for WSNs.

The data traffic pattern in a WSN highly depends on the characteristics of physical phenomena that it senses such as data duration, data rate, event priority, data size, data aggregation possibility, and frequency of sensed events. In some application, such as target monitoring which is a frequent operation in wireless multimedia sensor networks (WMSN), a sudden burst of bulk data is generated and should be routed to the sink. This may usually lead to congestion around the sink and/or some other intermediate nodes and should be handled without scarifying the network performance.

In recent years, both congestion control and avoidance have attracted great attention from the research community. Several protocols have been proposed for solving congestion control problem in WSN [1]. Among them, one could recognize four major categories: 1- data priority driven methods [2], 2-congestion avoidance methods [3], 3- rate control or traffic control [4], and 4-rerouting the traffic using alternative paths [5].

In this paper, a new routing algorithm is presented, which combines data packets using both inter and intra-network coding idea and avoids congestion areas (or nodes) in the network by employing alternative paths. The main advantage of the proposed algorithm is simultaneous control of network resources (via alternative paths) and traffic (via network coding) in order to improve network performance when facing the congestion. We have used hierarchical tree alternative path (HTAP) [6] as a resource control scheme to reroute traffic around congested area by creating dynamic alternative routes to the sink. Also, all traffic flows are randomly coded using linear intra-network coding [7]. Next, we use random linear inter-network coding to combine two flows with the same destination when they reach a shared network node. The paths of both flows are the same from this point toward the destination. Our detailed simulations confirm that combining resource control idea (i.e. HTAP) with a traffic control scheme (i.e. network coding) improves both packet delivery ratio and energy consumption rate compared to original HTAP.

Section 2 reviews available algorithms for congestion control. Section 3 presents the proposed algorithm in detail. Section 4 presents the simulation results and assesses the performance of the proposed method. Section 5 presents conclusion and some recommendations for future research.

## 2- Related works

Information produced in wireless sensor networks does not have a similar priority. That is, some data may be more important than other and should be delivered immediately. MCAR [2], which is a modified form of CAR (Congestion Aware Routing) algorithm, includes recent advances in MAC layer and a protocol for generating high priority routes. This algorithm is suitable for data resources with high mobility. A congestion control protocol with two specific characteristics has been presented in [2] which has been called SenTCP. The first characteristic is that it simultaneously uses both average packet service time and mean packet arrival time in order to estimate a sensor’s local congestion. As a second characteristic, it uses a step-by-step congestion control. In SenTCP, each intermediate node sends explicit backward feedback signal. The feedback signal, which carries local congestion degree and buffer's congestion rate, is used for regulating the sending rate of adjacent nodes. Using such feedback signals quickly remove congestion and decrease packet drop rate, which in turn leads to saving energy.

A congestion avoidance plan, named C3TCP, has been presented in [3]. This plan tries to resolve the problem of performance degradation in transfer layer protocols which itself has resulted from wireless network congestion. It has the ability to achieve higher efficiency by collecting useful information such as bandwidth and delay at the link layer.

In [5], a predictive congestion control protocol has been presented for WSNs, called decentralized, predictive congestion control (DPCC). When Transfer Control Protocol (TCP) is used as a transport protocol for wireless networks, it inevitably results in several packet drops, unfairness, performance degradation, and energy waste while attempting to resend packets. It predicts imminent congestion considering both queue utilization and a channel estimator algorithm. It then selects an appropriate rate for each active flow. An adaptive scheduling scheme is used to update packet weights and guarantee weighted fairness among active flows during congestion.

Topology-Aware Resource Adaptation (TARA) was presented to alleviate congestion in WSNs in [8] through increasing available resources by activating more sensor nodes. It uses an advanced capacity analysis model to estimate the capacity of produced topologies.

A resource control algorithm named HTAP has been presented in [6]. HTAP is a dynamic congestion control algorithm which decides to change routes based on local information such as congestion status of adjacent nodes. HTAP includes four steps: topology control, making a hierarchical tree, creating alternative routes, and managing low-energy nodes. In topology control phase, it tries to warrant the existence of the maximum possible routes by using minimum energy through data routing process. This is done by making a local minimum spanning tree for each node. The construction of the hierarchical tree is done when a node senses an event and wants to send the sensed data. It avoids congested and low energy nodes by using alternative routes in the hierarchical tree.

Network coding was presented to increase network throughput and reliability [9]. In network coding, incoming packets are stored and combined with each other before forwarding (store, combine, and forward paradigm). The Linear random combination of packets [7] ensures that other forwarding nodes, which hear ongoing transmissions, don't send identically coded packets. Therefore, the chance of receiving duplicate copies is reduced dramatically. MORE [10] is an opportunistic routing protocol which is independent of MAC layer and exploits intra-network coding. However, it doesn't consider combining packets from multiple flows. O3 is an optimized overlay- based opportunistic routing protocol which exploits opportunistic routing, inter-network coding, and rate restriction [11].

It builds an overlay network on top of existing wireless network. In the overlay network, overlay routing and inter-network coding are used. At underlay level, it uses opportunistic forwarding (i.e. intra-network coding) and rate restriction.

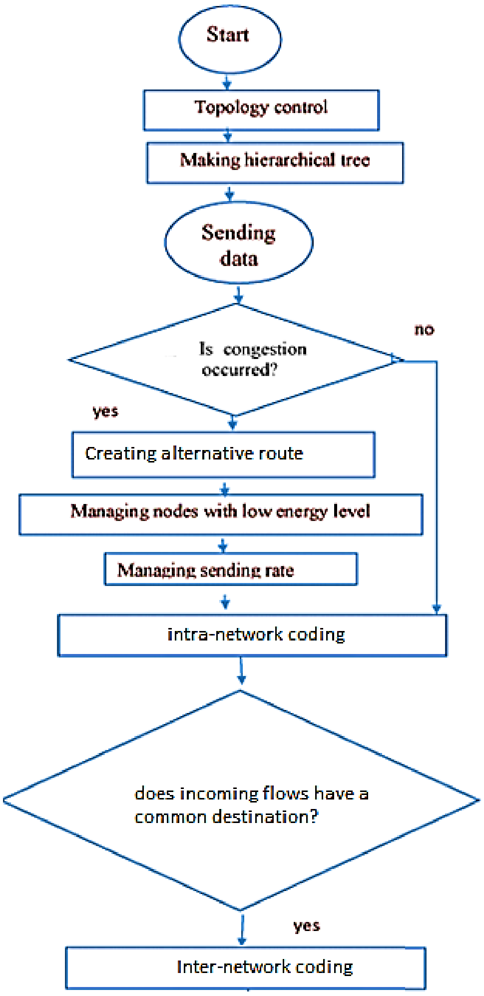


**Figure 1- MHTAP's flowchart**

## 3- MHTAP

The main goals of this study are 1-decreasing the number of ongoing transmissions, and improving overall power consumption in presence of congestion. We have made following assumptions about WSN. First, the senders are located in a specific region of WSN. Also, the sink node is located in the network corner and all sensors know about the sink location. Finally, all sensors have identical characteristics such as radio range, sending rate, and energy constraint. Our algorithm, which is called MHTAP for convenience, uses HTAP as a basis and inherits its simplicity and fast route selection. We decrease the number of transmissions by implementing network coding on top of HTAP. In fact, MHTAP improves the performance when facing congestion by controlling the traffic through network coding and making alternative routes (i.e. resources control) dynamically. Its flowchart is depicted in Fig. 1.

### 3-1- Topology control

The topology control phase of MHTAP is same as HTAP. Initially, topology control is implemented by using Local Minimum Spanning Tree (LMST) algorithm. In order to control topology more efficiently, the maximum degree of each node is set to 6 to prevent contention and interference by creating enough alternative routes. LMST is created in a distributed manner by all sensor nodes. Each individual node runs LMST algorithm independently.

### 3-2- Hierarchical tree construction

The hierarchical tree is constructed on top of topology control algorithm when a sensor node becomes data source after sensing an event. Like HTAP method, this step includes two main tasks: route creation and flow establishment. The route creation process is done by the source as follow. First, it set its level number equal to zero. It then sends the level discovery message to a selected set of its children. Upon receiving level discovery message by a child, it increases the level number and rebroadcasts the message. This process is continued until the message is delivered to the sink. The result of route creation process is multiple paths between the data source and the sink, which could be used later in the flow establishment phase. Each node in the constructed tree receives a weight based on its distance from the source and its remaining energy level. The congestion avoidance is carried out by extending the traffic through more resources (multiple paths) in the network. Also, MHTAP provides the possibility of using nodes with higher remaining energy level and lower congestion. It does not send data to congested nodes. The weight of the edge between node *i* and its child *j* in hierarchical tree is calculated as follow:

$$W_{ij}=\left(\frac{D_j}{E_j}\right) \tag{1}$$

Where $D_j$ is the position of the child node in relation to the sink (we assumed that each node knows its relative position to the sink) and $E_j$ is the child's remaining energy level. Downstream nodes piggyback the aforementioned parameters using ACK packets in response to successful reception of DATA packets. The ACK message is sent to an upstream node. The values of $D_j$ and $E_j$ parameters are normalized by the upstream node according to equation (2) before being used in equation (1).

$$N_i=\frac{n_i\text{-}n_{min}}{n_{max}\text{-}\ n_{min}} \tag{2}$$

#### 3-2-1- Alternative Path Creation

An alternative path is created by the congested node in order to control the congestion in the hierarchical tree. It is done when one of the following cases occurs. In each case, the node should have the ability to recognize and react to the occurring congestion.

- **Buffer-based congestion**: buffer-based congestion occurs when the reception rate of incoming traffic is more than the node's transmission ability.
- **Low energy:** it occurs when the node's energy level is less than minimum energy threshold, which is 5% of its initial energy level.
- Finally, if the sender could not reduce its sending rate, the congested node should reroute some of crossing flows.

Like HTAP method, MHTAP also uses a local Congestion Detection (CD) algorithm performed by each node. When a node detects the congestion, it sends a message to its upstream nodes, who use the node as a next-hop for forwarding DATA packets, and asks them to create an alternative route for their upcoming DATA packets. In response, the upstream node selects another tree node with appropriate weight. The selected node's level should be equal to the congested node's level. Also, it temporarily inactivates the congested node until the congestion is finished. By doing so, the congested node will no longer be used as a next-hop by the upstream node. After the congestion recovery, the congested node must send another message to its upstream node, which activates the congested node again.

#### 3-2-2- Low Energy Nodes

When the energy level of a node falls below the minimum energy threshold, it broadcasts a proper message to inform its neighbors. Upon reception of this message, the receiving neighbor inactivates the power exhausted node. It also creates an alternative route in the case that the power exhausted node is on an active DATA path.

#### 3-2-3- Controling Sending Rate

In addition to resource control strategy described above, MHTAP also uses a traffic control approach for congestion control purpose. The node that receives a DATA packet from an upstream node estimates the service time and calculates congestion degree according to equation (3):

$$C_i=\frac{Ts}{Nr} \tag{3}$$

In above equation, $T_s$ is service time and $N_r$ is the number of inquiry produced by the current node. Then, it calculates the normalized difference between its current congestion degree ($C_i$) and the moving average of $n$ previous samples of $C_i$ ($\overline{C}_i(n)$) as follow:

$$\alpha_i=\frac{C_i-\overline{C}_i(n)}{C_i} \qquad (4)$$

If $|\alpha_i| \geq \varepsilon$, where $\varepsilon$ is a predefined threshold, the downstream node sends $\alpha_i$ and $C_i$ to its upstream node, which in turn reduces its sending rate or uses another downstream node. The upstream node reduces its sending rate provided that the rate doesn't become lower than minimum sending rate limit. Otherwise, it selects another downstream node and reset its sending rate to its initial value. If $\alpha_i \leq -\varepsilon$, the downstream node is capable of handling more traffic. Therefore, the upstream node is allowed to increase the sending rate provided that it does not cross maxim sending rate limit. By using this model, the sending rate from the source to the sink is appropriately controlled, which in turn control the congestion. In addition to controlling the energy consumption rate in high sending rate, the variable sending rate improves the packet delivery ratio and decreases energy usage in large networks.

#### 3-2-4- Network Coding

The main difference between HTAP and MHTAP is the implementation of network coding. MHTAP uses both inter and intra-network coding concepts simultaneously. The source transmits the data packet using batch concept. Each data batch consists of 16 different packets. The source encodes data packets from similar batch using random linear network coding. It keeps transmitting encoded packets from current batch until the sink acknowledges the current batch. The sink is able to decode the current batch when it successfully received 16 independents coded packets. It then sends an ACK packet toward the sender to allow it moving to the next batch. The source rate is controlled by the downstream node the very same as HTAP.

Each intermediate node stores encoded packets in its local buffer. Whenever it successfully received an encoded packet, it randomly combines the stored packets. For example, suppose that a node received three encoded packets, namely P1, P2, and P3. It transmits $\alpha P1+\beta P2+\gamma P3$, where $\alpha$, $\beta$, and $\gamma$ are randomly generated 8-its length coefficients. All calculations are performed on a finite and prime Galois field of length 8 ($GF(2^8)$). This procedure is called intra-network coding and is used by other protocol such as MORE [10].

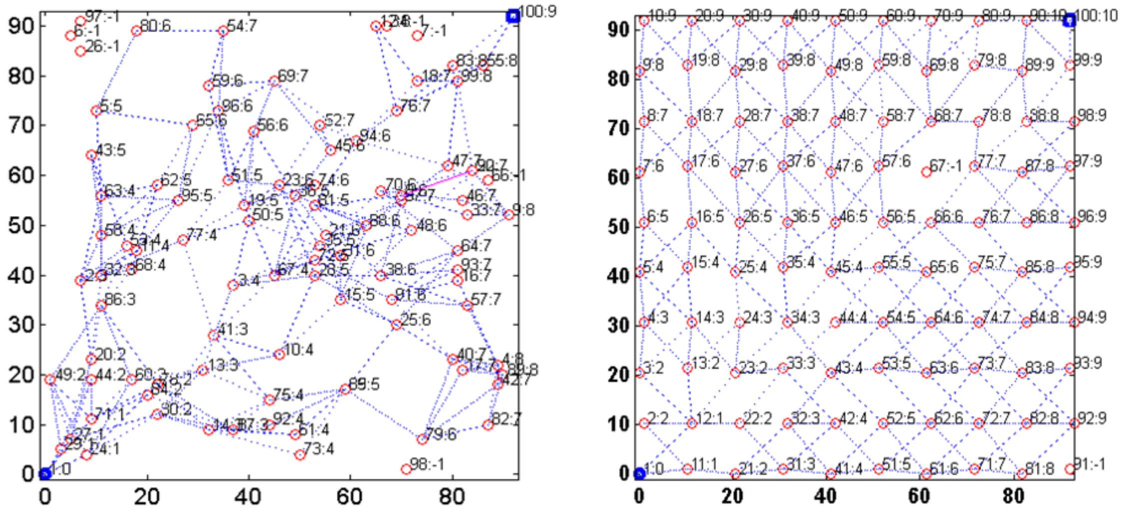
MHTAP has a unique feature. When two flows with the same destination crossing an intermediate node, the node performs inter-network coding. It doubles the batch size and produces randomly encoded packets composed of both flows. Downstream intermediate nodes act as if they process a single flow. In another word, they assume that the batch size is 32 and all packets belong to a single flow. The destination decodes the encoded packets when it received enough independent and encoded packets (i.e. 32). It then sends two ACK messages towards the sources of both flows. This will allow them to move to the next batch.

## 4- Simulation Results

We have implemented MHTAP and HTAP via MATLAB in order to evaluate their behavior comprehensively for different scenarios. Each sensor node is modeled by a Mica-z node, which its most important parameters are presented in table 1. In the conducted experiments, 100 to 400 sensor nodes are randomly placed in a 1000m×1000m environment. Network performance is examined in terms of packet delivery ratio (PDR) and total energy consumption (TEC) against the number of sensor nodes and source rate. We have used uniform distribution (Fig. 2a) and grid topology (Fig. 2b) for sensor nodes distribution.

**Table1- Simulation Parameters**

| Parameter | Value |
|---|---|
| Maximum Data Rate (Kbps) | 128 |
| Transmission Power (dbm) | 2 |
| Receive Threshod (dbm) | -74 |
| Transmission Current (mA) | 17.4 |
| Receive Current (mA) | 19.7 |
| Fragment Size (bit) | 1024 |
| Buffer Size (Bytes) | 128K |
| MAC Layer | CSMA/CA |

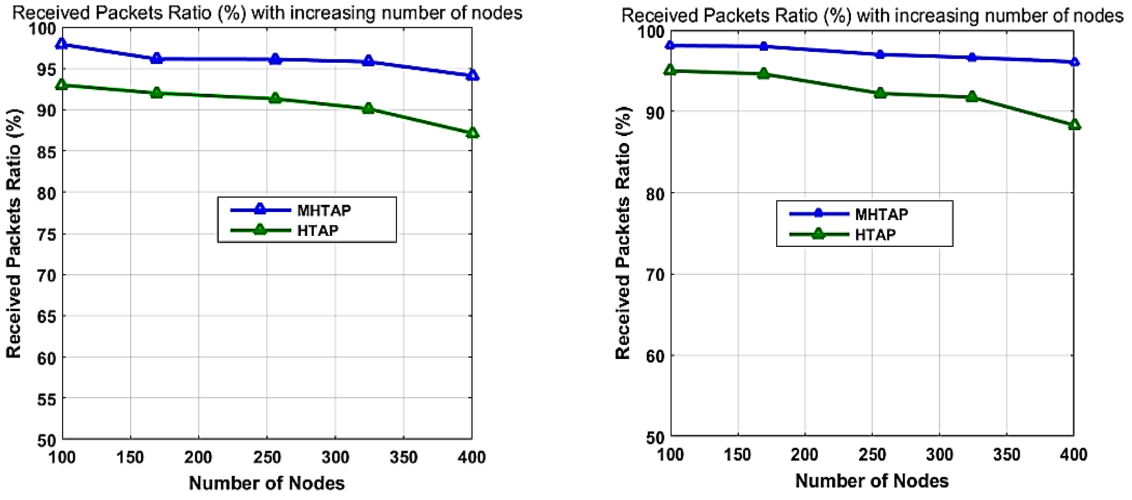


(a) (b)

**Figure 2- a) Uniform node distribution and b) grid topology**

### 4-1- Network Size

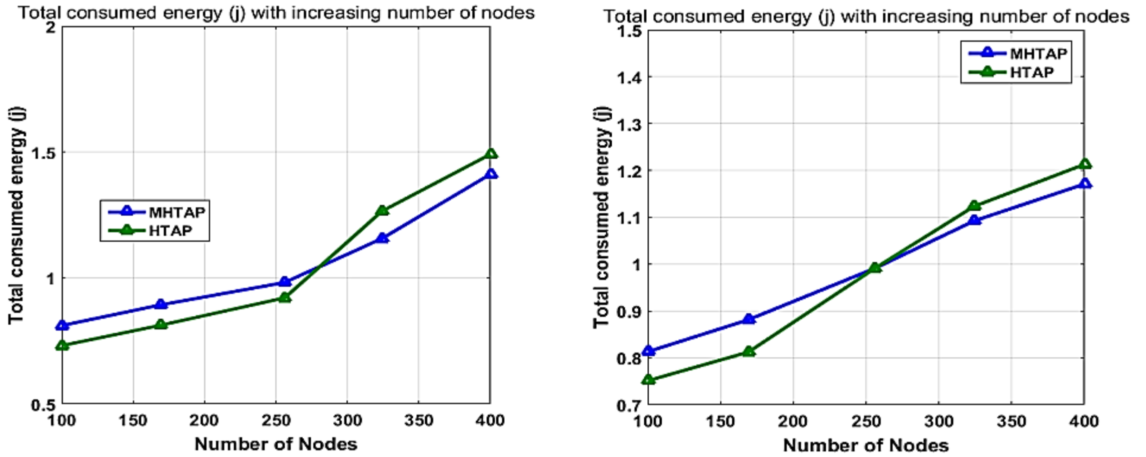
In these experiments, we aim to assess the scalability of MHTAP by examining its performance against network size. The number of nodes is varied from 100 to 400. It is assumed that each node is activated at different time intervals and sends its data packets using constant bit rate (CBR) traffic pattern. It is worth noting that the number of active flows is also proportional to the number of nodes. It should be mentioned that in these experiments, the field size is fixed. The network density therefore increases with the number of nodes. Fig. 3 shows the PDR for both uniform and grid topologies. As it is evident in the plot, delivery ratio slightly decreases with network size in both topologies for both methods. That is, an increment in nodes number increases the node density which in turn increases the collision probabilities between active flows. However, MHTAP always performs better than HTAP due to its inbuilt mechanisms. First, it uses intra network coding which increases the efficiency. Second, it combines individual flows when their destinations are the same.



(a) (b)

**Figure 3- PDR vs. network size; a) grid topology, b) uniform topology**

Fig. 4 depicts TEC vs. network size. Clearly, increasing the network size will increase the energy consumption of the network. But, MHTAP performs better than HTAP when the number of nodes is more than 250. That is, MHTAP performs better when there are more active flows in the network. In theses cases, it is possible to share more nodes and links between active flows. MHTAP uses this opportunity to reduce the network traffic and hence increase the delivery ratio. Also, both protocols perform better in uniform topology. The reason behind this behavior is that the average hop count and hence the average path length are lower in the uniform topology.

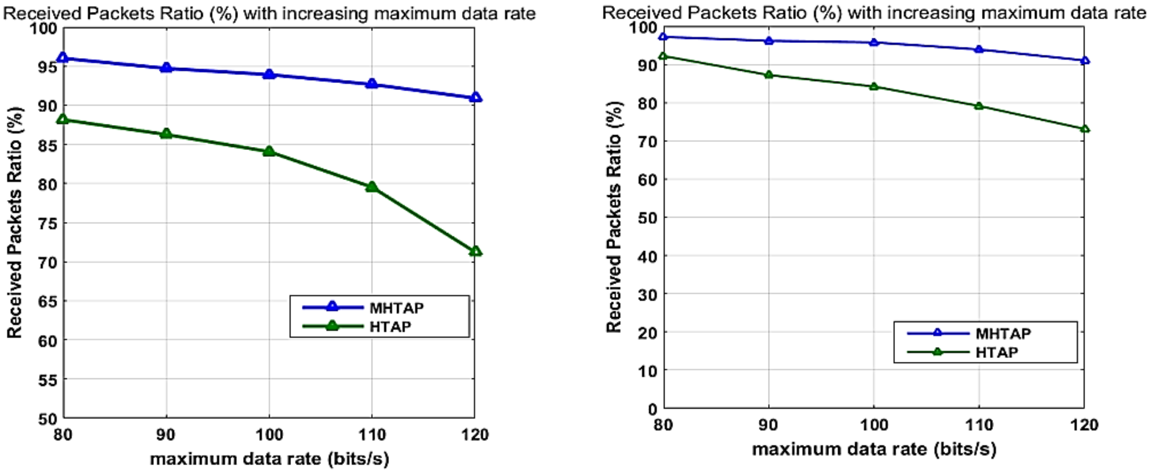


**(a)** **(b)**

**Figure 4- TEC vs. network size; a) grid topology, b) uniform topology**

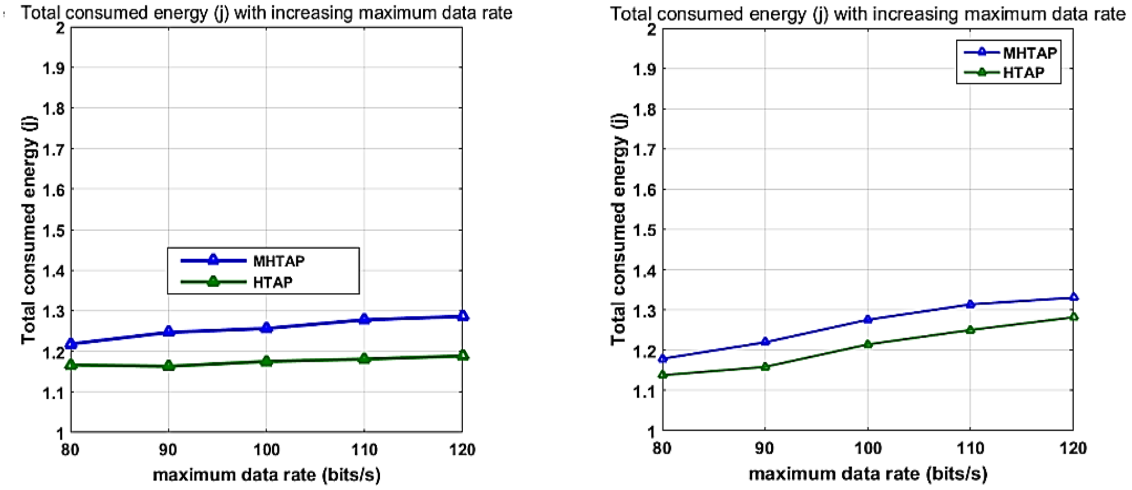
### 4-2- Source Rate

In these experiments, we examined MHTAP's performance for various source rates. As it is shown in Fig. 5, PDR decreases with source rate. Also, MHTAP outperforms HTAP since it decreases the number of required transmissions. When source rate increases, more intermediate nodes became congested. As a result, HTAP uses alternative paths, which further decreases PDR. In contrast, MHTAP decreases the traffic for intermediate nodes by using intra-network coding. Also, it combines active flows using inter-network coding to decreases the traffic. In Fig. 6, TEC is plotted vs. source rate. MHTAP consumes more energy than HTAP. But, it has used the energy for increasing PDR.



**(a)** **(b)**

**Figure 5- PDR vs. source rate. a) grid topology, b) uniform topology**



**(a)** **(b)**

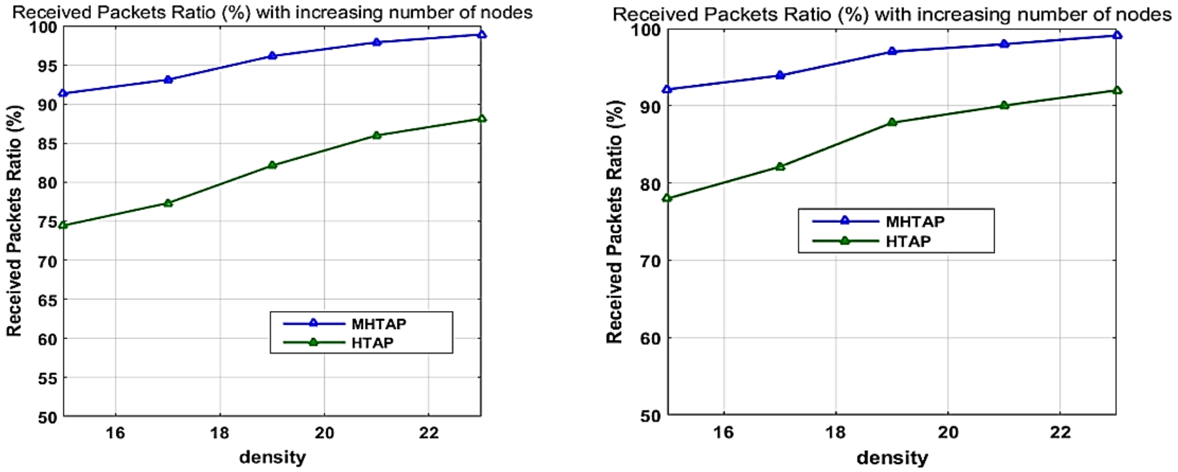
**Figure 6- TEC vs. source rate.a) grid topology, b) uniform topology**

### 4-3- Node Density

In this experiment, we aim to examine the effect of density changes on the performance of MHTAP. The network density is calculated as follow:

$$\text{Network\_density} = \frac{\text{Nodes\_Count} \times \pi \times r^2}{\text{Field\_Size}} \tag{5}$$

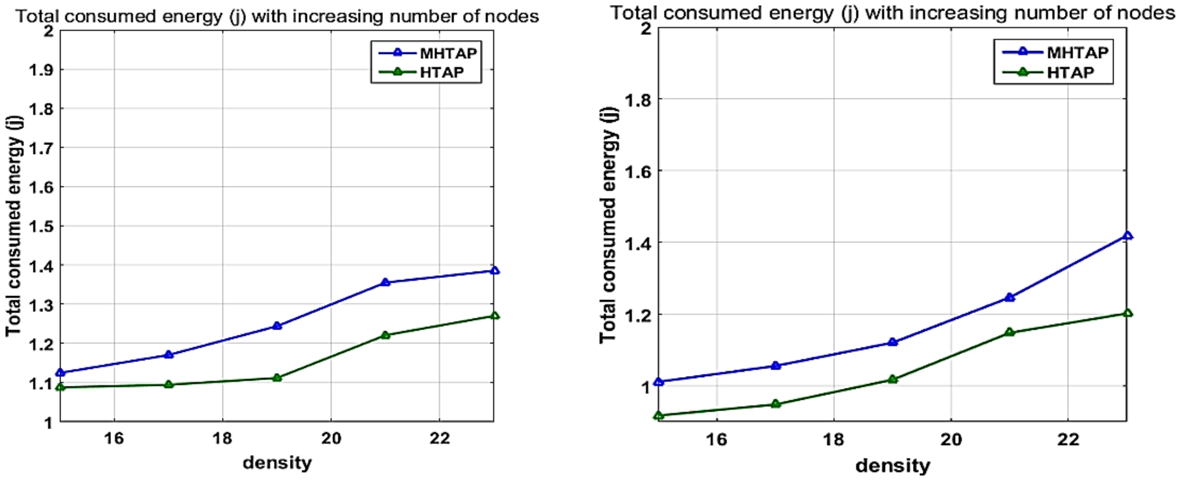
Where *r* is the radio range of a typical wireless node. In these sets of experiments, we have fixed the number of nodes. We adjust the network density by changing field size. Fig. 7 shows PDR vs. network density in the grid and random topologies.



**(a)** **(b)**

**Figure 7- Packet delivery ratio vs. network density. a) grid topology, b) uniform topology**

According to Fig. 7a, PDR increases with network density. It is worth noting that in Fig. 3, PDR decreases with network density. As said before, the number of active flows is proportional to the number of nodes. In Fig. 3, network density is increased due to increasing nodes count, which in turn increased the number of active flows. As a result, PDR decreases. But, in Fig. 7, the nodes count is fixed. Therefore, the number of active flows does not change. In fact, we must route a same amount of traffic in a denser network. Here, we expect that more neighbors exist per each wireless node. Also, each wireless path consists of short and reliable links. It has been proven that short and reliable links always performs better than long and unreliable links in wireless environments [6]. Finally, TEC is plotted vs. network density in Fig. 8. Again, we observe the same trend as previous plots.



**(a)** **(b)**

**Figure 8- TEC vs. network density a) grid topology, b) uniform topology**

### 5- Conclusion

In this paper, a new approach, called MHTAP, was presented in order to improve the routing process in WSN by combining network coding and resource control techniques. MHTAP utilizes inter and intra-network coding techniques as well as alternative route hierarchical tree simultaneously in order to control the congestion. This leads to dynamic control of congestion in intermediate nodes. MHTAP combine different flows which have some common steps toward their destination and decreases the number of required transmissions from the source to the destination. Therefore, MHTAP experiences better packet delivery rate than similar approaches. Our simulation results showed that MHTAP improved the packet delivery rate by 18.5% and 15.34% in the grid and random topologies, respectively, in comparison to HTAP. As a future work, optimization algorithms could be used for achieving a better balance between energy consumption and packet delivery rate.

## 6- References

1. Ghaffari, Ali. "Congestion control mechanisms in wireless sensor networks: A survey." Journal of network and computer applications, No. 52, pp.101-115, 2015.

2. Kumar, Raju, Crepaldi, Riccardo, Rowaihy, Hosam, F. Harris III, Albert, Cao, Guohong, Zorzi, Michele, and F. La Porta, Thomas. 2008. Mitigating Performance Degradation in Congested Sensor Networks. IEEE Transactions on Mobile Computing, 7 (6): pp. 682-697, DOI: 10.1109/TMC.2008.20.

3. Kliazovich, Dzmitry and Granelli, Fabrizio. 2006. Cross-layer congestion control in ad hoc wireless networks. Journal of the Ad Hoc Networks. (Nov. 2006), 4 (6): pp 687–708, http://dx.doi.org/10.1016/j.adhoc.2005.08.001.

4. Wu, Guowei, Xia, Feng, Yao, Lin, Zhang, Yan, and Zhu, Yanwei. 2011. A Hop-by-hop Cross-layer Congestion Control Scheme for Wireless Sensor Networks. Journal of Software, (Dec.2011), pp. 1-7, arXiv:1201.0207v1.

5. Zawodniok, Maciej and Jagannathan, Sarangapani. 2007. Predictive Congestion Control Protocol for Wireless Sensor Networks. IEEE Transactions on Wireless Communications, 6 (11): pp. 3955-3963, DOI: 10.1109/TWC.2007.051035.

6. Sergiou, Charalambos, Vassiliou, Vasos, and Paphitis, Aristodemos. 2013. Hierarchical Tree Alternative Path (HTAP) algorithm for congestion control in wireless sensor networks. Journal of the Ad Hoc Networks (January 2013), 11 (1): pp. 257-272, http://dx.doi.org/10.1016/j.adhoc.2012.05.010.

7. Ho, Tracey, et al. "A random linear network coding approach to multicast." IEEE Transactions on Information Theory 52.10 (2006): 4413-4430.

8. Kang, Jaewon , Zhang, Yanyong , and Nath, Badri . 2007. TARA: Topology-Aware Resource Adaptation to Alleviate Congestion in Sensor Networks. IEEE Transactions on Parallel and Distributed Systems, 18(7): pp. 919-931, DOI: 10.1109/TPDS.2007.1030.

9. Ahlswede, Rudolf, et al. "Network information flow." IEEE Transactions on information theory 46.4 (2000): 1204-1216.

10. Chachulski, Szymon, Jennings, Michael, Katti, Sachin, and Katabi, Dina. 2007. Trading Structure for Randomness in Wireless Opportunistic Routing. (SIGCOMM '07) Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communications, Kyoto, Japan (Aug. 27–31, 2007). ACM, New York, NY, 37 (4); pp. 169-180, doi> 10.1145/1282380.1282400.

11. Han, Mi Kyung, Bhartia, Apurv, Qiu, Lili, and Rozner, Eric. 2011. O3: Optimized Overlay-based Opportunistic Routing. MobiHoc '11 Proceedings of the Twelfth ACM International Symposium on Mobile Ad Hoc Networking and Computing, Paris, France (May 17-19, 2011) ,ACM, New York, NY, doi>10.1145/2107502.2107505.