A Novel Network Coding Scheme for Efficient Multicast and Unicast in Ad-hoc Networks

Mohamed Osama, Ahmed Shawish

***Abstract*—** Achieving higher throughput values in wireless ad-hoc networks is essential to support many applications. Network coding was found to be the most promising and effective approach for this purpose. However current network coding schemes, mainly relying on COPE, were able to improve the throughput gain in case of unicast flows only, while failing to achieve similar gain in case of the multicast scenario. With the notable flourish of conference-based and multimedia streaming applications that are mainly dependent on multicast flows, it became crucial to formulate a new network coding scheme that is able to handle both unicast and multicast flows with the same efficiency. In this paper, we introduce a novel network coding scheme to efficiently handle multicast flows simultaneously with the unicast ones with the same high efficiency and hence achieve a real high throughput gains. It even provides the option to favor one type of flow over the other when needed. The proposed scheme depends on graph theory to model packets and nodes in the network. A smart algorithm is then introduced to discover all feasible coding options in a way that smartly avoids the draw backs of previous schemes. Extensive simulation studies report the success of the proposed scheme to deliver nearly double of the COPE throughput gain in case of multicast flows while still delivering on par gain in unicast case.

***Keywords*—** Ad-Hoc networks, Multicast, Network Coding, Unicast.

I. Introduction

Minimal configuration and quick deployment make wireless networks (WLAN) suitable for last-mile Internet coverage. Specific characteristics of WLAN such as the broadcast nature, spatial diversity and packet redundancy give advantage to this type of networks over other candidates to support modern applications that require delivering of the same packet to multiple receivers. However, WLAN have been designed following same principles of wired networks where the protocols of wired networks have been grafted onto WLAN. The different nature of the wireless medium where most of the links are broadcasting, in contrast to unicast links found in wired networks, have caused conflict between the wired network design and the characteristics of the wireless medium. This conflict combined with limited bandwidth and resources make WLAN suffer low throughput.

Numerous methods have been proposed to increase the throughput and efficiently make use of the intrinsic characteristic of such environment [1]. Network Coding (NC) proposed in [2], is perceived as the most promising and innovative technique to increase the throughput of WLAN. NC manages to compress packets by exploiting the significant packet redundancy exhibited in WLAN, providing an increased flow of information per transmission, and thus achieving improved throughput. This is done by making use of the broadcast nature of wireless medium which amplifies the packet redundancy.

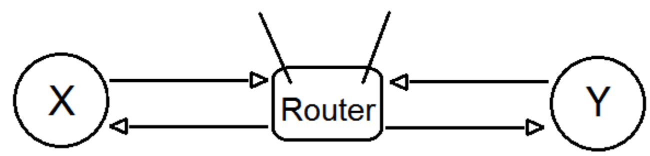

Fig. 1 A simple example demonstrating the throughput gain of using Network Coding

In Fig. 1, two nodes X and Y want to exchange a couple of packets through a router. Without NC, this would require 4 transmissions (i.e. X sends to router, router forwards to Y, Y sends to router and router forwards to X). Applying NC as in [2] reduces the number of transmissions to 3. The two nodes X and Y will transmit their respective packets to the router, which XORs the packets together in one packet and broadcasts the coded packet to the two nodes in one transmission. Each node recovers the packet intended for it, by XOR-ing the received coded packet with its own. This process exploits the packet redundancy in WLAN and makes use of the broadcast nature to deliver two packets in one transmission, and thus improves throughput.

COPE [3,4] which received warm reception from the research community and was considered the first practical scheme for NC demonstrated an efficient throughput gain in case of unicast traffic, while it didn't succeeded to provide a similar gain in the multicast case. Many follow up works like [5]-[9] all trying to improve the throughput in wireless networks. However, most of them focused on unicast without a similar attention to multicast. Even the works done targeting multicast only like [22] and [23] didn't report a considerable enhancement as it should be.

Recently, with the increasing demand of applications like all-informed voice, group push-to-talk, situational information sharing etc, supporting one-to-all and many-to-all (i.e.,

Mohamed Osama is with the Department of Scientific Computing, Faculty of Computer and Information Sciences, Ain-Shams University, Abbassia, Cairo, Egypt(mohamed.osama@fcis.asu.edu.eg)

Ahmed Shawish is with the Department of Scientific Computing, Faculty of Computer and Information Sciences, Ain-Shams University, Abbassia, Cairo, Egypt(ahmed.gawish@fcis.asu.edu.eg)

multicast) communication patterns in multihop wireless networks posed a problem that needed to be efficiently addressed. The need for an efficient scheme that is able to enhance the WLAN throughput in both multicast and unicast cases simultaneously became crucial.

Motivated by the absence of solutions that achieve better throughput gain for both unicast and multicast traffic, we propose a new enhanced network coding scheme, which we refer to as Graph-Based Network Coding "GBNC" that can handle both unicast and multicast flows simultaneously with the same throughput gain. GBNC also provides the option to tweak the scheme when the flow in the network is known to be majorly of a specific type through a novel approach which we refer to as the "Balance Switch". The problem of choosing the appropriate combination of packets to be encoded together is modeled using GBNC as a graph. Possible coding options are efficiently discovered and the one with the highest gain (i.e., number of new packets delivered to their destinations) is selected. The Balance Switch also provides different ways for calculating the coding gain if a specific type of flow is needed to be favored over the other. Because of the novel way of representing the problem, GBNC avoids the drawbacks of previous techniques and deals with multicast and/or unicast flow delivering equivalent throughput gain in both cases. Extensive simulation results show the ability of the GBNC to achieve similar throughput gain to that of COPE in unicast flows coming on top with a slight margin. GBNC also clearly outperforms COPE in case of multicasting by providing almost double or more the throughput gain.

The rest of this paper is organized as follows. Section II, gives a comprehensive background on the available network coding schemes. In section III, the proposed network coding scheme is addressed in details. In section IV, the simulations' results are presented and the achieved throughput gain is discussed. Finally, the paper is concluded and future work is listed in Section V.

II. BACKGROUND

The pioneering work on network coding started with a paper by Ahlswede et al. [2], who demonstrated that having routers encode different messages allows the communication to achieve multicast capacity. It was soon followed by the work of Li et al., who showed that, for multicast traffic (e.g., the butterfly scenario), linear codes are sufficient to achieve the maximum capacity bounds [10].

Koetter and M´edard [11] presented polynomial time algorithms for encoding and decoding, and these results were extended to random codes by Ho et al. [12]. However, all this work was primarily theoretical and assumed multicast traffic only. COPE [2,3], which attracted a lot of research interest, proposed the first practical scheme for one-hop NC across unicast sessions in wireless mesh networks [2]. Following papers tried to model and analyze COPE [13-15] and did not provide any considerable improvement.

Others proposed new coded wireless systems, based on the idea of COPE [16], [17]. In [18], the performance of COPE is improved by investigating its interaction with MAC fairness. Optimal scheduling and routing for COPE are considered in [13] and [15], respectively. I^2NC [19] built upon such work but did not handle multicast flow and focused on loss rate only. Use of network coding along with cooperative communication was found to provide throughput gain for TCP flow as in [20] but multicast flow was not considered and work was primarily for TCP and forced more complexity to incorporate cooperative communication.

Marium and Farhat [29] presented an approach that uses network coding to improve the performance and throughput. However the results do not improve the gains achieved by COPE or other approaches and does not handle the multicast scenarios. Piriya and Takuji [30] proposed a mechanism with network coding that is effective for allocating bandwidth to each user in wireless networks. However this mechanism deals with bandwidth auction case only.

Su-Kit and Dongyang [31] used multiple multicast trees in ad hoc networks to improve the throughput of multicast routing protocols. However, the technique presented uses conventional random coding scheme without any modification and did not handle the unicast flow scenarios.

MORE [21] is the first intra-flow NC-based protocol for reliable unicast and multicast over WMNs, in which nodes that overhear the transmission and are closer to the destination may participate in network coding and forwarding of the coded packets, forming forwarding belts toward the destinations. However belt forwarding can be inefficient, especially for multicast in which multiple overlapped belts are formed and many nodes intend to forward. Pacifier [22] improved upon MORE by using a multicast tree instead of multiple belts. Only nodes on the multicast tree are allowed to perform random NC. It is reported in [22] that Pacifier performs better than MORE for reliable multicast in WMNs. Both MORE and Pacifier relied on acknowledgments from the set of receivers and applied classic NC that is not suited for multicast flows. HoPCaster [23] outperformed Pacifier by integrating network coding and receiver-driven hop-to-hop transport to achieve high-throughput reliable multicast, yet it did not modify the coding scheme and did not handle unicast flow.

In [24] G. Lauer et al. propose Concerto; a RLNC-based broadcast protocol. The experimental evaluations reported in both [24] and [25] show that RLNC provides substantial coding gains/performance improvements in a real network. All these protocols are proposed for single-source scenarios. A source groups a set of consecutive packets into blocks called generations or batches (we will use generations as the common term). Coding operations are confined to packets belonging to the same generation. In case a simulation scenario includes multiple sources, the basic idea is then simply replicated – each source creates, independently, generations of coded packets. To the best of our knowledge, only few papers [26-28] explicitly addressed multi-source wireless broadcast. None of these investigated in depth whether cross-source coding (i.e., combining packets from potentially different sources) provides performance improvements, compared to repeating the single source

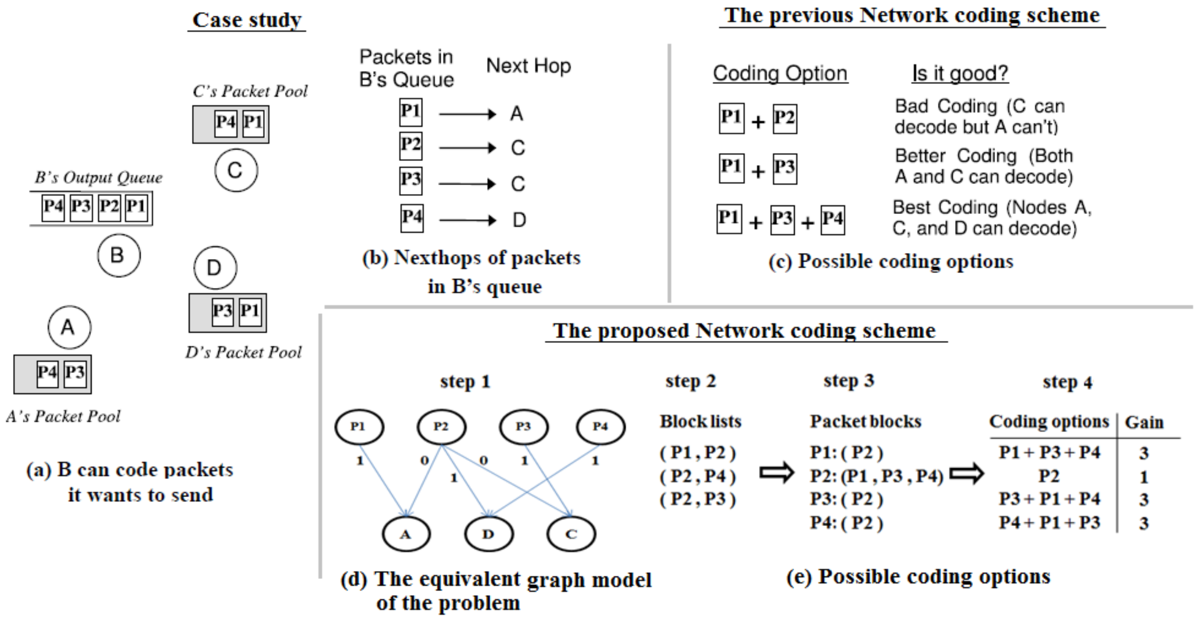

Fig. 2 Coding opportunity example under both previous and proposed Network Coding schemes

solution multiple times, once per source. In contrast to all the previous work, this paper proposes a new novel network coding scheme that handles unicast flow as efficient as COPE and outperforms it in case of multicast.

III. PROPOSED NETWORK CODING SCHEME

Applying NC helps to improve throughput by compressing the packets to be sent (i.e., through XOR-ing) and thus creating an increase in the amount of information transmitted per flow. However, current techniques and approaches that build on such concept as presented in COPE [1], haven't fully exploited the core of the problem. The focus of researchers is directed at increasing the coding opportunities, which is mostly done by assuming specific conditions or enforcing certain assumptions. Finding a way to actually improve the coding scheme, used for selecting packets to be coded together, would have a substantial effect on enhancing the throughput. Because such improvement is done on the basic level of NC, there's a chance of increasing the number of coded packets at each coding opportunity. Such increase provides current approaches the ability to achieve higher throughput and promises future trials of scalable gain since they all fundamentally depend on the coding step.

In order to come up with a new network coding scheme that is able to intuitively find better coding options, we incorporated graph theory in a novel way that is to the best of our knowledge, never done before. The transmitting node, its packets and neighbor hops, were all modeled as a directed weighted graph. This graphical representation takes the problem to a completely different domain, allowing better exploitation of the core nature of the problem through a new perspective. The graph representation is created following specific rules and coupled with an algorithm that explains the steps required to analyze the graph and reach a final solution.

To explain the proposed network coding scheme, let's consider the same case study addressed in COPE. In Fig. 2(a) the source node B has a set of 4 packets in its output queue (i.e., P1, P2, P3, and P4). Nodes A, D and C are randomly scattered neighbor nodes around B. Each neighbor node has overheard some of the packets. The next hop of each packet in B's output queue is shown in Fig. 2(b). Fig. 2(c) shows the coding options obtained from applying COPE [1] coding scheme. In Fig. 2(d) and (e), the 4 main steps forming the proposed network coding scheme are demonstrated. These steps start by converting the problem of finding the best selection of packets to be coded together into a graph problem and uses lists in a novel way to get the possible coding options with their corresponding gains in order to select the best one.

Step 1: Graphical model construction

For a node to determine the best coding option (i.e., packets to select from its output queue), packets and their corresponding next hops are modeled as nodes in a directed weighted graph. This graph serves as the foundation used for the steps to follow. For ex: Fig. 2(d) shows the graph that models the output queue packets of node B in Fig. 2(a) and their next hops. The resulted graph represents the packets and their next hopes, as graph nodes. Each packet in this graph is connected to its next hop(s) by a directed weighted edge. Graph construction is based on the following rules:

1) Each packet P_i in the output queue of the source node B is represented as a graph node, if and only if, it is a new packet to at least one of the neighbor hops (i.e. , it is not in the packet pool of a neighbor hop). Accordingly, all the packets (i.e., P1, P2, P3, and P4) are graphically represented as each of them is new to at least one of the neighbor hops.

2) Each neighbor network node is represented as a graph node, if and only if, there exists at least one packet in the source node's output queue that is new to such node. Accordingly, all neighbor nodes (i.e., A, D, and C) are represented as they all confine to this rule.
3) A directed weighted edge connects a packet graph node to a neighbor graph node if this packet is new to such neighbor. This is why there is no edge between P1 and both nodes C & D as they already have P1 in their packet pools.
4) The weight assigned to any given directed edge is 1, if the packet P_i (where i =1,2,..4), which is the edge source, needs to be routed to the neighbor node at the destination of the edge, either as a destination or a relay hop. Otherwise the weight of this edge is 0. Accordingly, the weight of the edge between P1 and node A is 1 as it is next hop of such packet, while the weight of the edge between P2 and node A is 0 as it is not the next hop of such packet.

Step 2: Block List

Towards the objective of finding feasible coding options, block lists are constructed. Each block list identifies a group of packets that cannot be coded together. Packets are grouped together if their corresponding packet graph nodes connect to the same neighbor graph node (i.e. they are new for the same neighbor hop). In order to create such block lists, each neighbor graph node having more than one incident edge is addressed apart. Accordingly, nodes A, C, and D in Fig. 2(d) will be addressed apart as they all have more than one incident edge. The block list is then created by combining the sources of edges (i.e., the packets) incident to such node. For example, In Fig. 2(e) under step 2, P1 and P2 are grouped in a block list as they are the sources of edges incident to node A, P2 and P4 are grouped in a block list as they are the sources of edges incident to node D, and similarly for P2 and P3.

A reduction phase is then conducted over the final block lists. This reduction process iterates upon the block lists with the goal of removing groups that may be completely contained in another block list. For example if the block lists contain both (P1,P2) & (P1,P2,P4) the first block list (P1,P2) will be removed from the final lists as a block list is completely contained inside another one.

Graph Nodes and edges construction procedure

GraphNodes = {}
for Packet i=1 to M **do**
  Pick packet p_i
  for Neighbor j=1 to N **do**
    Pick neighbor n_j
    if $p_i \notin$ packet pool of n_j **then**
      if p_i routed to n_j **then**
        Add edge from p_i to n_j with gain 1
      else
        Add edge from p_i to n_j with gain 0
      end if
      if $n_j \in$ GraphNodes **then**
        Add edge between p_i and n_j
      else
        GraphNodes = GraphNodes ∪ $\{n_j\}$
        Add edge between p_i and n_j
      end if
    end if
  end for
  if $\exists j$: $p_i \notin$ packet pool of n_j **then**
    GraphNodes = GraphNodes ∪ $\{p_i\}$
  end if
end for

BlockingList construction procedure

BlockingList = Φ
for NeighborGraphNode i=1 to N **do**
  Pick neighbor n_i
  Blocking = Φ
  if n_i count of edges >1 **then**
    for Edge j=1 to M **do**
      Pick edge e_j
      Pick packetnode p source of e_j
      Blocking = Blocking ∪ p
    end for
    BlockingList = BlockingList ∪ Blocking
  end if
end for

Step 3: Packet Blocks

This step aims at creating per packet block list named as packet blocks. Here, each packet is carefully addressed with the help of the block list to identify the list of other packets that can not be coded with it. For example, P2 can not be coded with P1, P3, and P4 as the block list includes (P1,P2), (P2,P3) and (P2,P4). Hence the packet blocks report this fact as P2: (P1, P3, P4).

Step 4: Coding option & gain

In this step, each packet is carefully addressed a part as a candidate for selection to be the first packet in the coding option, the packet is picked if and only if there does not exist a packet in its packet block that has higher gain than it. If the packet can not be selected the algorithm simply considers another packet, if it can be selected the remaining packets are examined for selection as long as they are not blocked according to the packet blocks of the already selected packets and they do not block a packet with higher gain.

The total gain of each computed coding option is calculated as the sum of gains of the packets selected in such coding option. The best coding option is selected based on the highest gain supplied. Fig. 2(e) shows a table demonstrating each feasible coding option computed and its total calculated gain.

Codingways computing procedure

codingways = Φ
for PacketNodes i=1 to M **do**
  codingway = Φ
  currentblocked = Φ
  Pick packetnode p_i
  if p_i has highest gain in its blocklist **then**
    codingway = codingway ∪ p_i
    currentblocked = currentblocked ∪ packets in

```
      p_i blocklist
      for PacketNodes j=1 to N do
         if j ≠ i & p_j ∉ currentblocked & ∄ packet p
         in p_j blocklist: (p ∉ currentblocked & gain of
         p > gain of p_j) then
            codingway = codingway ∪ p_j
     currentblocked = currentblocked ∪
            packets in p_j blocklist
         end if
      end for
      codingways = codingways ∪ codinway
   end if
end for
si=1
for codingways i=2 to M do
   Pick codingway c_i
   Pick codinway s of index si
   if c_i gain > s gain then
      si= i
   end if
end for
```

Further in depth analysis and simulation led to even more optimization that can be made to achieve higher throughput gains in wider range of scenarios and topologies making the scheme more dynamic and efficient. Due to the amplification of packet redundancy caused by the broadcast nature of WLAN, a case of what we call a dummy listener may arise. In such case, any neighbor node in the range of the transmitting node can block the coding of packets, if such node hasn't overheard any or most of the packets being coded. The blocking node can be considered dummy (i.e., packets do not need to be decoded at such node), if none of the packets being checked for encoding together, needs to be routed to that node (i.e., the edge connecting the packets and such node has weight of 0). Even though it might not seem of big importance, such case might be the cause of blocking an otherwise feasible coding option of packets. Since the topology of WLAN varies greatly and can't be expected, in some topologies or scenarios such case can be the cause of significant throughput drop and decrease in both the number of packets to be coded together and the number of opportunities of coding.

Fig. 3 shows the simplest case of a dummy listener scenario. On the left in the figure, node A has 2 packets P1 and P2 in its output queue. Neighbor node B has overheard packet P1, neighbor node D has overheard P2 while neighbor node C has not overheard any packet (i.e., acting as the dummy listener in this case).
The next hop of packet P1 is node D and that of packet P2 is B. The graph model representing such topology is shown on the right in the same figure. Applying the proposed scheme without any modification will not find any feasible coding option as P1 and P2 block each other at the dummy listener node C. However by applying the modification, later described, that addresses the dummy listener scenario this problem is solved.

A. Packet Blocks revisited

A modification is applied in this step to create the packets blocks taking into consideration the dummy listener case. As shown in Fig 3, node C does not have any of the packets yet none of the packets is actually routed to such node.In such scenario both COPE and unmodified GBNC fail to provide any network coding gain as no feasible coding option is found. Further processing on the block lists of each packet is done to remove the blocked packets resulting from dummy listener. For example the block list of P1 is (P2) but by carefully addressing apart each packet in the list, it is found that both P1 and P2 are not actualy intended to be routed to the node causing the blocking since their edges incident to node C have a value of zero . Therefore the block lists are reduced removing the packet blocks caused by dummy listener nodes. In the scenario shown in Fig. 3 double the throughput is achieved as the required number of transmission will be reduced from 2 to 1.

In real life scenarios and actual topologies, packet flow in the network may happen to be mostly multicast flow or unicast flow. Thus having the ability to fine tweak the coding scheme to be optimized to give advantage to a flow type without sacrificing the other type is crucial in supporting as diverse type of applications and scenarios as possible.

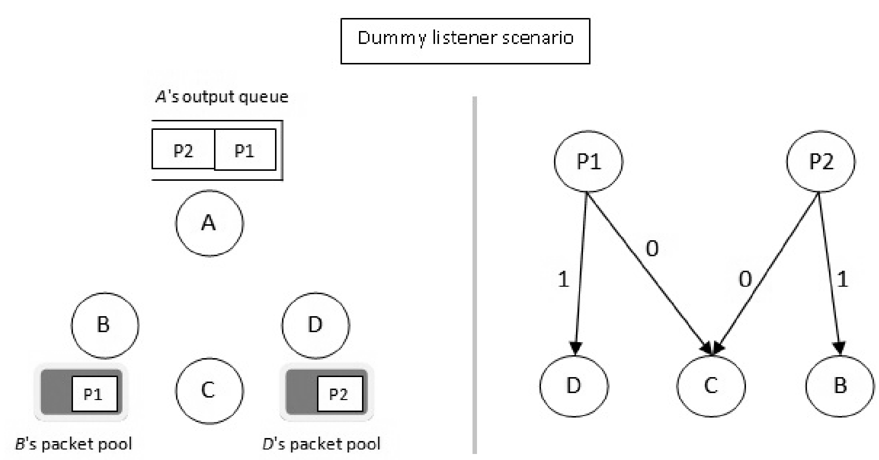

Fig. 3 A simple topology example with its equivalent graph representation

B. Balance switch

Another concern that stands out and can affect throughput is whether there is more room to optimize and achieve higher throughput, when information about the type of flow in the network is known or can be expected. In order to provide more optimization for specific network scenarios where the majority of flow in the network is known to be primarily of one type either unicast or multicast, we introduce the idea of what we call a balance switch. Through three different modes of calculating the total gain of a computed coding option, the balance switch provides the ability to alter the way the total gain of a coding option is calculated. This determines the type of flow that will be favored over the other. The three balance switch modes are as follows:

1-Balanced mode: The sum of packets' gains selected in a coding option is cosidered the gain of such coding option

2-Unicast-favor mode: The gains of multicasted packets in the coding option is calculated as 1 giving advantage to coding options with more unicast packets

3-Multicast-favor mode: Coding options having higher number of multicasted are ranked higher than those with the same total gain but less number of multicast packets thus giving advantage to multicast flow.

These modes provide further optimization to make the scheme more adaptive for specific scenarios or applications.

IV. Results and Discussion

In this section, we demonstrate the results obtained from extensive simulation experiments comparing the new proposed optimized network coding scheme to COPE.

The topology used for simulation consists of 18 randomly placed static Ad-hoc nodes with randomly picked source-destinations pairs , uniform arrival and normal distribution for packet arrival. Wireless medium channel transmission rates of 6,8,...,22 was used in simulation by varrying the arrival rates of the packets. UDP packet size was set to 80 bytes conforming to the G711 voice codec. We verified our implementation by simulating unicast flows using both no-network-coding and COPE. Each simulation results present the average of the conducted simulation trials.

As depicted in Fig. 4 the results obtained for the simulated COPE scheme are similar to those obtained in [2,3]. Fig. 4 shows that the throughput gain obtained by the proposed scheme is almost identical to the gain obtained by COPE when the flow type in the network is unicast only. The figure plots the aggregate end-to-end throughput as a function of the demands, with GBNC, with GBNC handling dummy listener, with COPE and without any scheme. Without any coding applied, throughput starts to deteriorate as the demands increase because of the effect of higher contention levels and consequent loss of packets induced by collisions. Applying coding reduces the number of packet transmissions resulting in higher level of throughput. GBNC achieves almost identical throughput gain to that of COPE. This proves that in case of unicast flow which COPE mainly addresses, GBNC manages to provide the same gain. Taking into consideration the dummy listener case, GBNC manages to detect more coding opportunities which translates to higher throughput gain.

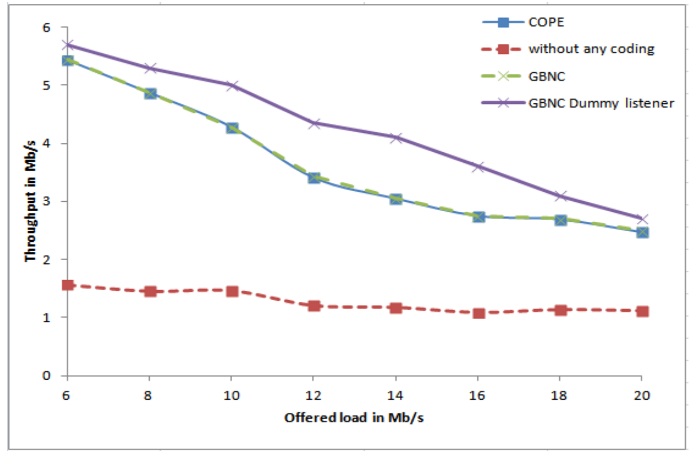

Fig. 4 Throughput gain against flow rate for unicast case

Fig. 5 shows the simulation results of multicast flow along with unicast flow. As depicted from the figure, COPE yields almost the same throughput as with no coding since multicasted packets will block the selection of other packets. GBNC manages to provide nearly similar throughput gain as in the unicast flow case. This is because of the way GBNC is designed. It combines unicast packets along with multicast packets. Accounting for the fact that multicast packets need to be routed to more than one hop, makes GBNC able to select the coding option with the highest packet delivery gain. This leads to delivering the maximum number of new packets to their intended hops based on the gain of each coding as illustrated in section III.

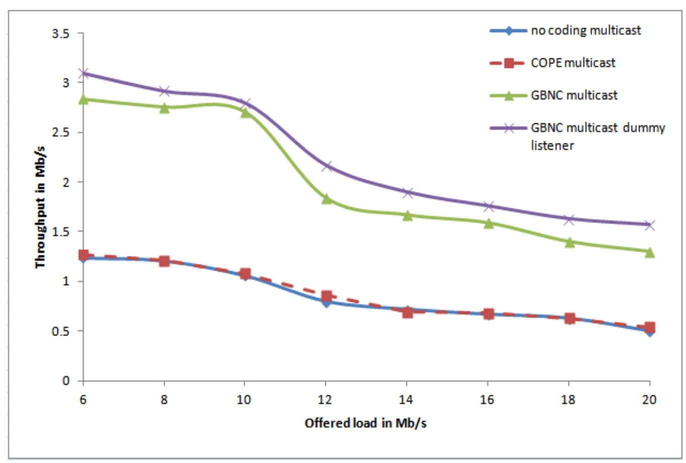

Fig. 5 Throughput gain against flow rate for multicast case

The proposed multicast GBNC scheme achieves nearly double or more the throughput gain compared to COPE. Due to marking the edges in our modeled graph by 0 or 1 according to whether the packet is routed to a specific node or not, GBNC efficiently comes up with the best coding selection of packets to deliver the maximum number of new packets in every transmission where COPE scheme fails as new packets blocks all other packets if selected and no better selection is considered. Handling the dummy listener case gives GBNC more advantage. More coding opportunities are spotted and higher gains from existing opportunities are achieved.

In Fig. 6 we demonstrate the effect of selecting different modes for the balance switch. The figure shows the average throughput value calculated for multiple percentages of unicast flow in the network (ranging from 50 to 100). As depicted from the figure, setting the balance switch to Unicast-Favor mode gives better throughput values especially between the percentages of 60 and 80. When the unicast flow is favored at these values, there is higher number of affected coding options that has more unicast packets in it.

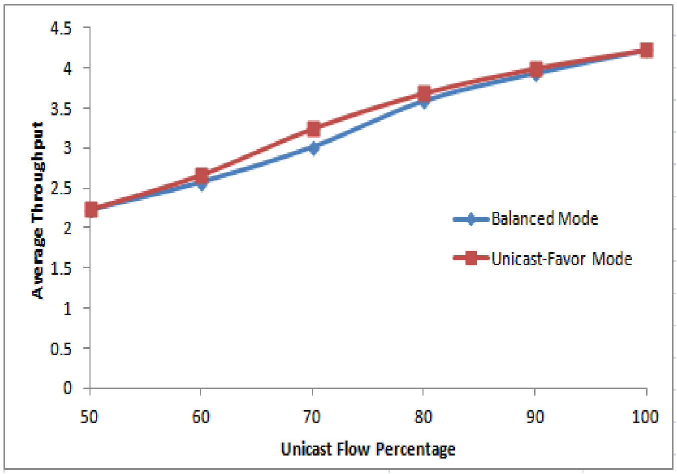

Fig. 6 Average Throughput value for different percentages of unicast flow

As the percentage goes down near 50 or up to 100 this effect starts to diminish. As the percentage of unicast flow approaches 50 percent, there are another 50 percent of multicast flow packets that gets affected negatively by the favoring of unicast flow packets. This leads to more delayed or dropped multicast flow packets causing a drop in the overall throughput that counters the gain caused from favoring the unicast flow. Also when the percentage of unicast flow approaches 100 percent, almost all the flow in the network is unicast thus there is no added gain from the Unicast-Favor mode since all the packets in the network are of the same type. It should be noted that the exact percentage at which the effect of the balance switch is highest, varies according to the topology and routes of the flows inside the network causing coding opportunities between multicast and unicast packets to arise. The effect of the Multicast-Favor mode is expected to be the same in case of increasing the percentage of multicast flow caused by the delayed or dropped unicast packets.

V. CONCLUSION AND FUTURE WORK

This paper proposed a new scheme of network coding to efficiently encoding both unicast and multicast traffic simultaneously. The scheme incorporated the graph theory to model packets and nodes in the network with the help of a developed smart algorithm to discover all feasible coding options in a way that efficiently avoided the draw backs of previous schemes. The proposed technique had also provided the option to tweak the scheme when the flow in the network is known to be majorly of a specific type through a novel approach which we refer to as the "Balance Switch".

The extensive simulation studies report the ability of the proposed scheme to achieve double of the famous COPE's throughput gain in case of multicast flows, while still able to handle the unicast traffic with the same performance reported by the COPE. These results introduce a significant enhancement on the network coding solution which is the basis of many works aiming at improving the network's throughput. In addition, it presents a step forward towards building a set of novel coding schemes that are independent of the flow type.

ACKNOWLEDGMENT

The authors wish to thank Prof. Essam Khalifa and Prof. Hossam Faheem at Ain-Shams university for their in depth discussions and valuable comments.

REFERENCES

[1] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: a survey," *Computer Networks*, vol. 47, no. 4, pp. 445 – 487, March 2005.

[2] R. Ahlswede, N. Cai, S.-Y. Li, and R. Yeung, "Network information flow," *IEEE Trans. Inform. Theory*, vol. 46, pp. 1204–1216, Jul. 2000.

[3] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft,"XORs in The Air: Practical Wireless Network Coding," in *ACM SIGCOMM*, 2006

[4] S. Katti, H. Rahul, W. Hu, D. Katabi, M.Medard, and J.Crowcroft, "Xors in the air: Practical wireless network coding," *IEEE/ACM Transactions on Networking*, vol. 16, pp. 497–510, Jun. 2008.

[5] P. Chaporkar and A. Proutiere, "Adaptive Network Coding and Scheduling for Maximizing Throughput in Wireless Networks," in *ACM MOBICOM*, 2007.

[6] S. Rayanchu, S. Sen, J. Wu, S. Banerjee, and S. Sengupta, "Loss-Aware Network Coding for Unicast Wireless Sessions: Design, Implementation, and Performance Evaluation," in *ACM SIGMETRICS*, 2008.

[7] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading Structure for Randomness in Wireless Opportunistic Routing," in *ACM SIGCOMM*, 2007.

[8] B. Scheuermann, W. Hu, and J. Crowcroft, "Near-Optimal Co-ordinated Coding in Wireless Multihop Networks ," in *ACM CONEXT*, 2007.

[9] Georgios S. Paschos, Constantinos Fragiadakis, Leonidas Georgiadis and Leandros Tassiulas, "Wireless network coding with partial overhearing information, " in *IEEE INFOCOM*, 2013.

[10] S. R. Li, R. W. Yeung, and N. Cai. Linear network coding. In *IEEE Transactions on Information Theory*, 2003.

[11] R. Koetter and M. M´edard. An algebraic approach to network coding *IEEE/ACM Transactions on Networking, 2003*.

[12] T. Ho, R. Koetter, M. M´edard, D. Karger, and M. Effros. The Benefits of Coding over Routing in a Randomized Setting. In *ISIT, 2003*.

[13] P. Chaporkar and A. Proutiere, "Adaptive network coding and scheduling for maximizing througput in wireless networks," *in Proc. of ACM Mobicom*, Montreal, Canada, Sep. 2007.

[14] J. Le, J. Lui, and D. M. Chiu, "How many packets can we encode? - an analysis of practical wireless network coding," *in Proc. of Infocom*, Phoenix, AZ, April 2008.

[15] S. Sengupta, S. Rayanchu, and S. Banarjee, "An analysis of wireless network coding for unicast sessions: the case for coding-aware routing," *in Proc. of Infocom*, Anchorage, AK, May 2007.

[16] Q. Dong, J. Wu, W. Hu, and J. Crowcroft, "Practical network coding in wireless networks," *in Proc. of MobiCom*, Montreal, Canada, Sept. 2007.

[17] S. Omiwade, R. Zheng, and C. Hua. "Butterflies in the mesh: lightweight localized wireless network coding," *in Proc. of NetCod*, Lausanne, Switzerland, Jan. 2008.

[18] F. Zhao and M. Medard, "On analyzing and improving COPE performance," *in Proc. of ITA*, San Diego, CA, Feb. 2010.

[19] Hulya Seferoglu, Athina Markopoulou, K. K. Ramakrishnan, "Intra- and Inter-Session Network Coding in Wireless Networks", *Infocom*, pp 1035-1043, 2011.

[20] Yifei Wei, Mei Song and F. Richard Yu, "TCP Performance Improvement in Wireless Networks with Cooperative Communications and Network Coding", *ICC* 2012.

[21] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing", *ACM SIGCOMM* 2007.

[22] D. Koutsonikolas, Y. Liu, C. Wang, "Pacifier: high-through, reliable multicast without crying babies in wireless mesh networks," *INFOCOM* 2009.

[23] Rami Halloush, Hang Liu, Lijun Dong, Mingquan Wu and Hayder Radha, HopCaster: A Network Coding-Based Hop-by-Hop Reliable Multicast Protocol, *Globecom* 2012

[24] G. Lauer, and D.S. Morris, "Network coding performance: An emulation experiment," 2008 *Military Comm. Conference,*pp.1-7, Nov. 2008.

[25] V.Firoiu et al., "Experiences with Network Coding within MANET Field Experiments," *2010 Military Comm. Conference*, pp. 1239-1244, Nov. 2010.

[26] C. Fragouli, J. Widmer, and J.-Y. L. Boudec, "A network coding approach to energy efficient broadcasting: from theory to practice", *IEEE Conference on Computer Communications*, pages 1-11, Apr. 2006

[27] C. Fragouli, J. Widmer, and J. Le Boudec. 2008. "Efficient broadcasting using network coding", *IEEE/ACM Trans. Netw.*, Apr. 2008, pp. 450-463.

[28] Y. Kondo, H. Yomo, S. Yamaguchi, P. Davis, R. Miura, S. Obana, "Reliable Wireless Broadcast with Random Network Coding for Real-Time Applications", *Conference on Wireless Communications and Networking,* pp. 1-6, 2009

[29] Marium Jalal Chaudhry, Farhat Saleemi, Fatima Jalal Chaudhry " Network Coding: An Excellent Approach for Overloaded Communication Era." *WSEAS Transaction on Communications*, 2009.

[30] Piriya, Takuji " VCG Auction-based Bandwidth Allocation with Network Coding in Wireless Networks" *10th WSEAS international conference on Applied computer and applied computational science*, 2011

[31] Su-Kit and Dongyang " Improving Performance of Multicast Routing in Ad Hoc Wireless Networks" *9th WSEAS International Conference on Applied Informatics and Communication*, 2009

Mohamed Osama received his B.Sc. degree in Computer Science in 2008 and his premaster in Computer Science in 2009 both from Ain Shams University, Cairo, Egypt. In 2009 he worked as a researcher developing Arabic search engines and in 2010 he worked as an R&D engineer developing distributed web based framework for ERP. He is currently filling the post of a teaching assistant in the department of Scientific Computing, faculty of Computer & Information Sciences, Ain Shams University, Egypt. His research cover wireless networks, network coding, distributed computing, graph theory and sequence alignment.

Dr. Ahmed Shawish received his B.Sc. degree in Computer Science in 1997 and received his M.Sc. degree in Computer Science in 2002, both from Ain Shams University, Cairo, Egypt. In 2009, he got his Ph.D. degree from Tohoku University, Japan. He is currently assistant professor in the Scientific Computing department, faculty of Computer & Information Sciences, Ain Shams University, Egypt. His researches cover supporting VoIP applications over wired & wireless networks, network coding, and Ad-Hoc routing protocols. Currently, he is focusing on Cloud Computing. He has published over 30 referred technical papers in these areas. He is member of IEEE.