

Network Coding Via Opportunistic Forwarding in Wireless Mesh Networks

Jian Zhang
ECE Department
Rutgers University
New Brunswick, U.S.A.
jianz@caip.rutgers.edu

Yuanzhu Peter Chen
Department of Computer Science
Memorial University of Newfoundland
St. John's, Canada
yzchen@cs.mun.ca

Ivan Marsic
ECE Department
Rutgers University
New Brunswick, U.S.A.
marsic@caip.rutgers.edu

Abstract — Network coding has been used to increase transportation capabilities in wireless mesh networks. In mesh networks, the coding opportunities depend on the co-location of multiple traffic flows. With fixed routes given by a routing protocol, the coding opportunities are limited. This paper presents a new protocol called BEND, which combines the features of network coding and opportunistic forwarding in 802.11-based mesh networks to create more coding opportunities in the network. Taking advantage of redundancy of packets among the forwarder candidates, our protocol bends the routes locally and dynamically to attain better coding opportunities. This higher coding gain is verified using a network simulator.

Keywords - network coding; opportunistic forwarding; 802.11; mesh network

I. INTRODUCTION

Network coding is an innovative technique for improving potential network throughput and robustness [4]. It changes the way that packets have been treated for a long time in that they can be combined when transported [1]. COPE [6][5] presented the first practical implementation of network coding to achieve higher unicast throughput in multi-hop wireless networks. The basic idea of how network coding can improve unicast is illustrated in Fig. 1. In Fig. 1(a), suppose that node X has a packet $P1$ for node O , and that node O has a packet $P2$ for node Y . Assume that both of these packets must be relayed by node C . In addition, we assume that X and Y are within each other's transmission range. Without network coding, four transmissions are needed to achieve such a message exchange. By coding, however, it can be done with only three transmissions, as follows. After X transmits $P1$ and O transmits $P2$, forwarder C broadcasts a packet P_x obtained by XOR-ing $P1$ and $P2$. Assume that $P1$ has been overheard by Y when it is transmitted by X . Upon receiving P_x , both receivers O and Y decode P_x by XOR-ing it with $P2$ on O and $P1$ on Y , respectively. Similar coding gain can be achieved in the example of Fig. 1(b), where node X has a packet $P1$ for node O and node D has a packet $P2$ for node Y . The

Fig. 1(a) scenario, referred to as “Y-topology”, is typical around mesh gateways, while the Fig. 1(b) “X-topology” can be seen in general traffic patterns. The authors of COPE also provided other scenarios, like cross and wheel topologies, where higher coding gains can be achieved. Apparently, these scenarios do not occur as often as the X- and Y-topologies. Therefore, in this paper, we investigate the possibility of creating more coding opportunities using those two topologies.

In COPE, the paths of flows are decided by the routing protocol used. Hence, the coding opportunities depend on the occurrences of X/Y-topologies formed in the paths. Coding in this case is limited by the fixed-path routing. In the scenario shown in Fig. 2(a), suppose that node O is a mesh gateway and all other nodes are mesh stations. Assume that an inbound stream is routed from X via A to O and that an outbound stream is routed from O via B to Y (the gray arrows). Even if network coding is allowed, there is no such an opportunity because these flows are link-disjoint. A similar situation is in Fig. 2(b), where four transmissions are still needed for one exchange even with COPE.

One proposed solution to the above problem is coding-aware routing [9][8][7], which takes coding opportunities into account while calculating path metrics. This approach requires not only the knowledge of network topology but also of the global information of current traffic distributions, which may change very quickly and is hard to maintain up to date. A more practical option is adaptation based on local knowledge,

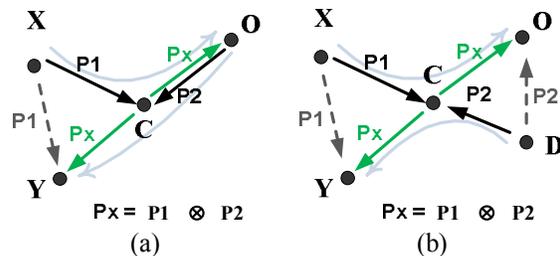


Figure 1: Coding scenarios

rather than global route planning. That is, to deviate locally from the intended intermediate node and select another forwarder where coding is possible. In Figs. 2(a) and 2(b), when node B is selected as an alternative forwarder for the flows in both directions, the typical X/Y-topologies are created for coding. More interestingly, we notice that not only node B but essentially any station within the range of X , Y , and O can be a potential forwarder. Thus, it is unnecessary to use a dedicated forwarder. Moreover, we find that the “ubiquitous”-forwarder strategy can significantly increase coding opportunities. Fig. 2(c) illustrates the basic idea. When O transmits packet $P2$ to Y via the intended forwarder, say D , other nodes B and C overhear it due to the broadcasting nature of wireless communication. Likewise, more than one copy of $P1$ may exist in the network after being transmitted by X . Essentially, any intermediate node overhearing both $P1$ and $P2$ and within transmission range of X , Y and O can perform encoding and forwarding. The more there are such potential forwarders, the higher the chance that coding can be achieved. Even with packet loss due to a channel error, such as $P1$ at B in the figure, the redundancy of packets in the network, when properly utilized, will leave some nodes, like node C here, available for coding.

The strategy previewed above is how BEND, proposed in this paper, creates more coding opportunities in the network. It works as the ray of light bends in presence of a gravitational field, thus named *BEND*. Here, the gravity is the coding chance at the forwarders. As indicated by the solid black arrows in Fig. 2, our strategy “bends” the routes prescribed by the routing protocol, switching dynamically among the candidate forwarders to utilize coding more frequently. Such method of not sticking to fixed paths and using redundancy of packets is similar to opportunistic forwarding as in [3]. In this paper, we show that the coding ratio, thus capacity, in a mesh network can be greatly improved by BEND.

The rest of the paper is organized as follows. We start off in Section II by a speculation on conscious use

of the broadcasting nature of wireless signaling as an innovation in the research of data communications, which leads to the proposal of BEND. In Section III, we describe the unique set of features of BEND and the challenges in order to implement these features. Then we provide the design of BEND in Section IV. Therein, we detail how to create more coding opportunities just around the mobile stations near a mesh gateway. The higher coding gain achieved by these creations is verified by the experimental studies presented in Section V. In Section VI, we further discuss the advantages of BEND in the context of some related work before the concluding remarks Section VII.

II. EXPLOITING WIRELESS BROADCAST NATURE

Research on wireless networking has mostly been focusing on improving the communication capabilities of wireless links. Wireless link untethers users and devices from the wired network backbones and enables mobility. To achieve this, shortcomings of wireless channels are overcome to provide reliable and high-bandwidth capacities.

Most of the research has focused on curbing the interference caused by the broadcast nature of wireless signaling. An example of battling this nature is the adoption of virtual carrier sensing via the coordination of the sending and receiving sides [2]. However, recently there has been a realization that this broadcast nature can be exploited to tap on unrealized advantages of wireless signaling.

Some attempts to achieve high unicast performance by utilizing such a broadcasting nature have been made more recently. Out of those, the two notable, although dissimilar, proposals are ExOR [3] and COPE [6]. In ExOR, similar to anycasting any neighbor en route can forward an overheard data packet as long as it is determined that such an opportunistic forwarding gets the packet closer to its destination. Such an innovation enables a packet to travel by long hops incidentally when the channel condition is good, and ensures the transmission using closer neighbors. The effectiveness of this approach has been tested on the MIT RoofNet

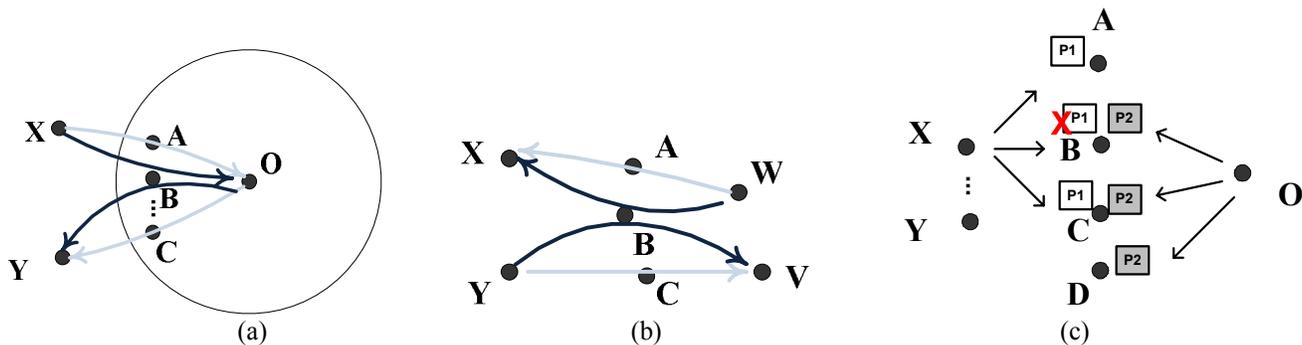


Figure 2: BEND

[7]. Network coding indicates that information flow can be split and combined to increase network capacity. The operation of splitting and combination in this case is made possible by the broadcasting nature of wireless links. As the first practical network coding for unicast traffic, COPE has been engineered by way of sniffing the wireless channel and coding multiple packets so that a single transmission can benefit multiple receivers. Such studies on exploiting wireless link's broadcasting nature reveal a new hope in this area.

The advantage of wireless broadcasting is not fully achieved by COPE alone. COPE relies on traditional routing to find paths between sources and destinations. Since the routes are fixed and only one path is used at a time, the availability of coding-enabling topologies is limited and thus the coding potential in the network is not fully exploited. By BEND, we overcome the above limit by introducing *opportunistic forwarding* for the purpose of coding. In BEND, all nodes operate in promiscuous mode and are allowed to "help" forwarding a packet. Unlike ExOR, which prioritizes forwarders by their distances to the destination, BEND favors those forwarders with a chance to transmit coded packets. As our simulation results show, this significantly enriches the coding opportunities in the network.

III. FEATURES AND CHALLENGES OF BEND

BEND has a unique set of features in its design:

- BEND is the first attempt to combine opportunistic forwarding and practical network coding in wireless networks. On one hand, as an enhancement of COPE, BEND creates more coding opportunities to achieve higher capacity. On the other hand, with opportunistic forwarding it employs the geographic diversity of different potential forwarders for robustness and redundancy, which are important for mitigating packet losses due to channel errors.
- BEND unifies the needs of traffic separation and concentration dynamically. In multi-hop wireless networks, traffic flows should be separated to minimize the interference among them. Conversely, for network coding to function, the same traffic flows should be co-located. These two needs were difficult to balance with traditional methods before BEND. One innovation of BEND is in providing a way to handle these two tendencies adaptively.
- BEND can be used both for the typical last-2-hops scenario and any general end-to-end scenario.
- BEND is mostly a MAC layer solution, and it works independently of any source or link-state routing protocol. Distance vector based protocols need slight modifications as described below. BEND is based on 802.11 MAC and is easy to implement for practical uses.

In order to implement the above features, we must address the following challenges:

Maximizing coding chances—To promote coded transmissions for throughput gain, a mechanism is needed to ensure that the packets have a better chance to be coded and transmitted by one forwarder than to be transmitted native. This must be handled without starving any flows or nodes in the network.

Coding conditions—When a node has a packet to forward, it needs to know if coding it with another backlogged packet may save bandwidth. This means, to find out if the receivers can decode the packets, once coded. It is challenging to acquire and maintain the information of which neighbors have which packets for decoding.

Reliable link-layer broadcast—Since a coded packet is intended to multiple receivers, an efficient and reliable link layer broadcasting mechanism is needed as a building block.

Duplication of packets—All nodes operate in the promiscuous mode for opportunistic forwarding. As a result, a packet may be overheard and queued at multiple neighbors. There must be a mechanism to make sure that it is forwarded by only one of these neighbors.

Storage of overheard packets—The preemptive nature of BEND requires a node to temporarily store a potentially large number of packets for coding. Thus, there must be an effective way to manage these packets. Which overheard packets should be queued for forwarding? Which need to be buffered for decoding? And when and how they should be discarded for the purposes of saving storage and avoiding duplication?

These challenges are addressed in the next section where details of BEND are presented.

IV. DESIGN

In this section, we present the major components in the design of BEND.

A. Header specification

BEND performs coding and tagging at MAC layer. It requires a modification to the DATA and ACK headers of the existing 802.11 MAC Specifications.

In Fig. 3 we highlight the fields modified or added for BEND. The header of DATA may have a different format depending on if the payload is native or encoded. If native, besides the sender address (SA) and receiver address (RA), the header includes the IP address of 2nd next hop (described in Section IV-B below). This information is passed down from the network layer when the transmission function is invoked. If encoded, it has the two receiver addresses in RA-1 and RA-2, and packet IDs (PID) for both native packets. The packet ID is generated by creating a 4-byte hash value out of the source IP and sequence number carried by the IP packet, as in COPE. A 2-bit type field in the frame control specifies frame types, i.e., native DATA, encoded DATA, ACK or other 802.11 frame.

Each ACK contains an SA and the packet ID of the native packet to acknowledge. Notice that BEND uses SA in ACK instead of RA as in the 802.11 Specifications. The reason is described in Section IV-C.

B. 2nd next hop en route

BEND has a pre-emptive nature with a *helper node* (B in Fig. 2) forwarding DATA frames on behalf of an *intended forwarder* (A or C). When a node requests help from its neighbors to forward a packet, it finds the IP address of the 2nd next hop (denoted by **2NH** in the rest of the paper) along the path to the destination. Then, it sets the 2nd-next-hop field in MAC header and transmits this DATA frame. For example, in Fig. 2, node X includes node O 's address when transmitting packet $P1$. Similarly, for the other flow, node O sets $P2$'s 2NH as Y . The knowledge of 2NH for X or O is provided by the routing module used in the network. If a source or link-state routing is used, this is trivial. But such knowledge is not immediately available for distance-vector based routing protocols. However, 2NH information can be obtained by minor modifications to distance-vector protocols. We simply add a "via" field to each distance vector in routing packets. That is, in the routing table broadcast to neighbors, each entry destination is associated with distance estimation plus via which neighbor this distance is established.

Upon receiving a DATA frame, only neighbors which are one-hop away from the 2NH specified in the frame header are allowed to forward it, either native or coded with other queued packets. This guarantees that the packet propagation is restricted within a "band" along the route without flooding the network.

C. To code or not

When a packet $P1$ is received by a potential forwarder, either intended or not, it sends it up to network layer. The network layer fills in the next-hop and TTL fields and sends it down. Then, BEND searches among the queued packets to find a coding partner $P2$ for $P1$. The following matching criteria help to find the partner by ensuring that their corresponding receivers can decode the coded packet:

1. The 2NH of $P1$ is $P2$'s previous forwarder, or one of its neighbors;
2. The 2NH of $P2$ is $P1$'s previous forwarder, or

one of its neighbors.

For example, in Fig. 2(c), $P1$ and $P2$ are queued at node C . $P1$'s 2NH is set by node X to O , which happens to be the previous forwarder of $P2$. And $P2$'s 2NH is Y , which is $P1$'s previous forwarder, X 's neighbor. Thus, when $P1$ and $P2$ are encoded as Px , O can decode it by XOR-ing Px with $P2$, and Y decodes it with $P1$, which was overheard from X previously.

If the above conditions are satisfied, those two packets are marked as ready for encoding.

D. Storage of the overheard packets

A node in BEND uses three data structures to store, in the native form, the packets that it 1) needs to forward encoded, 2) needs to transmit as native, or 3) overhears or originates, which can be used for the future decoding.

Two adaptive queues, Q1 and Q2, are used to store the packets for the first two cases, respectively. Packets stored in Q1 and Q2 will either be transmitted when scheduled or removed if the node is certain that a downstream neighbor has received it. In particular, the packets satisfying above coding criteria are placed together in Q1, the encoding queue. Otherwise, if no partner is found, the received packet is stored with other packets originated by this node in Q2, the native queue. A hash table is also generated for packets in both queues to facilitate retrieving packets quickly by using packet ID. This reduces processing time when removing a packet which is ACKed by the next-hop receivers.

Each node keeps all packets for case 3) in a temporary buffer for a period of time. When the buffer overflows, it drops oldest packets. Similarly, a hash table is used to quickly find the corresponding packet for decoding whose ID is specified in the MAC header of the received encoded DATA frame.

E. Two-level prioritization

Since a coded transmission doubles the link utilization, BEND gives it a higher priority in scheduling. This is achieved at two levels, i.e. within a node and among a set of contending nodes.

In a loaded network, the end-to-end delay is dominated by the queuing delay at individual nodes. This is particularly true for nodes located near a mesh router. Since the coding opportunity is transient, BEND is designed to seize these opportunities effectively.

MAC header of native data frame	Frame control	Duration	RA	TA	BSSID	Seq-ctl	2 nd next hop		
MAC header of encoded frame	Frame control	Duration	RA-1	TA	BSSID	Seq-ctl	RA-2	PktID-1	PktID-2
MAC header of ACK	Frame control	Duration	SA	FCS	PktID				

Figure 3: Mac headers of BEND

Within a forwarding node, the encoding queue (Q1) is assigned a higher priority weight than the native queue (Q2) (0.8 and 0.2, respectively, in our settings). The forwarder generates a random number uniformly between 0 and 1. If the number is greater than 0.2, the node schedules an encoded packet. Otherwise, it schedules a native packet. With these tunable weights, BEND gives encoded packets better chances for transmission and yet does not starve the native packets.

To achieve higher coding ratio, it is also important to prioritize forwarders that have encoded packets ready to send in MAC, since their native copies could also be transmitted native by other forwarders. 802.11 MAC regulates that, after a node decides to send a data packet, it must back off for a time interval uniformly distributed between 0 and $cw-1$, where cw is a changing window size to coordinate contending nodes. cw is initially 32 and is doubled every time a transmission attempt fails. BEND revises this by using an initial value of 16 for encoded packets, instead, so that they have a better chance to win the channel over a native packet.

When packets for encoding are scheduled, they are coded by XOR-ing and the result is encapsulated with a MAC header for encoded frame (Fig. 3). Their packet IDs and receiver addresses are also in the header.

F. Reliable broadcast and duplication avoidance

When a packet P_x coded from P_1 and P_2 arrives at a receiver, say Y in Fig. 2(c), Y searches the buffer for P_1 by its ID indicated by RA-1 or RA-2 in the header of P_x . P_1 should have been overheard earlier when transmitted by X and buffered. If P_1 is found, Y successfully decodes P_2 from P_x and continues to send an ACK.

A coded packet needs to be broadcast to two receivers and the link layer is responsible for the reliability of the broadcast. The 802.11 Specification only includes an unacknowledged, and thus unreliable, broadcast. Previous work, such as COPE, resorts to an

approximate reliability. Here, we devise a reliable link-layer broadcast. In essence, both receivers of a broadcast packet are polled by the sender in the order as specified in the BEND header. Thus, the receivers send their ACKs back-to-back to the sender without collision.

The ACK contains the address of the sender (instead of the receiver as in regular 802.11) and the ID of the received native data packet. The rationale for replacing the receiver's address with the sender's address in ACK is that, in BEND, an ACK is used to free all copies of a packet at all forwarders. When these forwarders receive the ACK, they check the corresponding packet in the queue using packet ID in the ACK frame. If the ACK's sender (SA) is the next-hop node of the data packet, which means that the packet has already been successfully received by its next-hop receiver, this data packet can be removed to avoid duplication.

V. SIMULATION

We test BEND on three scenarios by network simulation (NS2) and compare its performance with 802.11 original, 802.11 with opportunistic forwarding, and 802.11 with fixed-path coding, i.e., COPE. The nodes in the scenarios are located in three tiers. The first tier is the mesh gateway, the second consists of 2 or 3 forwarders, and the third consists of 4 or 8 end nodes. The combinations are denoted by 1-2-4, 1-2-8, and 1-3-8. Between each 3rd-tier station and the mesh gateway, there are an upstream and a downstream constant bit rate flows with packet size set to 1000 bytes. The NS2 PHY layer model is enhanced with realistic channel propagation with random frame losses based on SINR.

Fig. 4(top) shows the aggregate throughput gain of BEND and other schemes over the original 802.11 for scenarios 1-2-4, 1-2-8 and 1-3-8. We can observe that BEND outperforms 802.11 original by 82%, 95% and 82%, respectively. Note that the coding gain of BEND is more than a simple addition of the individual gains of opportunistic forwarding and the COPE employed separately. This is attributed to more coding opportunities by BEND. The total numbers of successfully decoded packets, Fig. 4(bottom), verifies this. With opportunistic forwarding, BEND transports 46%, 71% and 37% more coded packets than COPE.

From our tests for some ad-hoc END-TO-END scenarios, such as 4-4-4, 4-2-4 and 8-2-8, BEND also shows a promising improvement over COPE. Some more general scenarios with longer-path traffics are currently under testing.

VI. DISCUSSION AND RELATED WORK

Traditional routing protocols' obliviousness to the coding opportunities was noticed in [9][8]. Their solutions focus on routing in the network layer. Such attempts are usually referred to as coding-aware routing. The idea is to compute routes for given flows in a

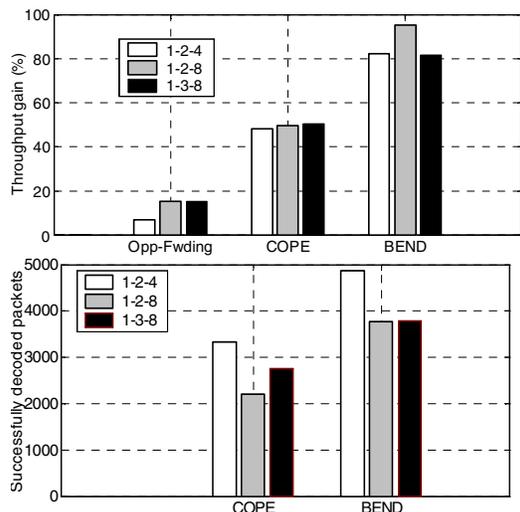


Figure 4: Performance of BEND

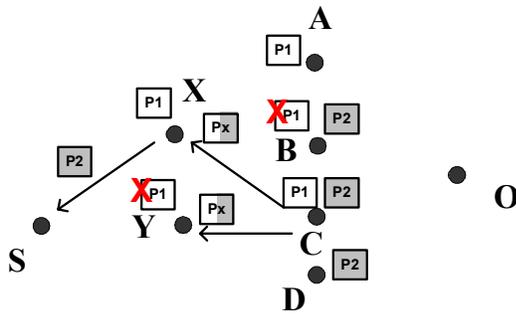


Figure 5: Another level of redundancy

network, taking network coding gain into account, so that the expected total number of transmissions needed to transport the flows is minimized. This is of great importance in theory but the distributed implementation can be rather involved. To compute coding-aware paths, each node needs to maintain global information of all the flows in the network. The time granularities of traffic lifetime and route update period are usually discrepant. The calculated routes will typically be long dated before being applied to the flows used for the route calculation. More so, due to the extremely close coupling among these flows, any unilateral change of route adopted by an intermediate node will invalidate the purpose of the global routing metric, i.e. number of transmissions. Moreover, coding-aware routing approach is still based on traditional routing with a single fixed path for each source/destination pair and the redundancy of packets in the network cannot be utilized.

BEND aims at achieving a high coding ratio for each stage of forwarding. It is not globally optimal, but it is flexible, adaptive and practically effective. It only requires local information and the implementation overhead is low. Since it is per-packet decision for coding, as opposed to per-flow path adaptation, it is more agile to the dynamics of traffic. In addition, BEND takes advantage of packet redundancy in the network by opportunistic forwarding. With a set of multiple potential forwarders instead of one, the coding chances are greatly improved. Moreover, coding-aware routing needs to consider not only the coding gain by combining traffic flows but also their consequential interference. These two needs have been difficult to balance with traditional methods with fixed-path routing. Unlike this, in BEND all potential forwarders form a “band” along the routing path. If the bands of two flows overlap, the overlapping forwarders can perform coding when the coding conditions are met, e.g., when flows are heading in opposite directions. Conversely, if coding is not possible, e.g., parallel flows in the same direction, the disjoint forwarders in their respective bands can transmit these packets in parallel. The latter effect is similar to ExOR’s ability of bypassing congested nodes.

Current implementation of BEND only makes use of redundancy of packets for encoding. The performance of

BEND can be further improved by exploiting the redundancy of packets for decoding. Take an example similar to the one in Fig. 2(c). Here, in Fig. 5, P_2 is originated by O and destined to S via intended forwarder Y . The early state is the same as before where C encodes P_1 and P_2 and relays the coded P_x to Y and O . Suppose that, unfortunately, Y lost P_1 transmitted previously by X . Without P_1 , Y is unable to decode P_x and it has to be retransmitted in current BEND implementation. But, thanks to the broadcasting nature again, P_x is overheard by X which has P_1 available in its buffer. Thus, node X takes over and forwards the decoded P_2 to S . The decoding success ratio is therefore improved.

VII. CONCLUSION

Broadcasting can cause interference in mesh networks, but it also brings the benefit of facilitating network coding. When applied efficiently, network coding will significantly improve the network’s transportation capabilities. The BEND protocol proposed here, starts with the goal of creating more network coding opportunities with a low overhead. It averts the impasse of possibly scarce coding opportunities as with COPE. The key of BEND is to create more coding chances by bending the routes only locally and dynamically via opportunistic forwarding. BEND can be used for the typical last-2-hop scenario in mesh networks. This last-2-hop area around mesh gateway is usually heavily-loaded, which necessitates an enhancement of node capabilities, but it is also dense with nodes, which provides an arena to exercise BEND. BEND can also be used in end-to-end scenarios. Our simulation results indicate significant throughput gain and coding ratio increase by BEND

REFERENCES

- [1] R. Ahlswede, N. Cai, S.R. Li, and R. W. Yeung, “Network information flow,” *IEEE Trans on Information Theory*, 2000.
- [2] V. Bharghavan, S. Demers, S. Shenker, and L. Zhang, “MACAW: A media access protocol for wireless LANs,” *ACM SIGCOMM*, 1994.
- [3] S. Biswas and R. Morris, “Opportunistic routing in multi-hop wireless networks,” *ACM SIGCOMM*, 2005.
- [4] C. Fragouli, J-Y Le Boudec, and J. Widmer, “Network coding: an instant primer,” *ACM SIGCOMM Computer Communication Review*, 36(1), pp. 63-68, 2006.
- [5] S. Katti, D. Katabi, W. Hu, R. Hariharan, and M. Medard, “The Importance of Being Opportunistic: Practical Network Coding For Wireless Environments,” In *Proc. 43rd Allerton Conf. on Communication, Control, and Computing*, Monticello, IL 2005.
- [6] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, “XORs in The Air: Practical Wireless Network Coding,” *ACM SIGCOMM* 2006.
- [7] MIT Roofnet, <http://pdos.csail.mit.edu/roofnet/doku.php>
- [8] B. Ni, N. Santhapuri, Z. Zhong, and S. Nelakuditi, “Routing with Opportunistically Coded Exchanges in Wireless Mesh Networks,” *Poster session of IEEE SECON* 2006, <http://arena.cse.sc.edu/papers/rocx.secon06.pdf>
- [9] S. Sengupta, S. Rayanchu, and S. Banerjee, “An Analysis of Wireless Network Coding for Unicast Sessions: The Case for Coding -Aware Routing,” *IEEE INFOCOM*, 2007.