Distributed opportunistic and diffused coding with multiple decoders in wireless mesh networks

Thuong Van Vu, Thi-Mai-Trang Nguyen, Guy Pujolle

Distributed Opportunistic and Diffused Coding with Multiple Decoders in Wireless Mesh Networks

Thuong Van Vu
University of Pierre and Marie Curie
Laboratory of Computer Science of Paris 6
4 place Jussieu, 75005 Paris, France
thuong-van.vu@lip6.fr

Thi Mai Trang Nguyen
University of Pierre and Marie Curie
Laboratory of Computer Science of Paris 6
4 place Jussieu, 75005 Paris, France
thi-mai-trang.nguyen@lip6.fr

Guy Pujolle
University of Pierre and Marie Curie
Laboratory of Computer Science of Paris 6
4 place Jussieu, 75005 Paris, France
guy.pujolle@lip6.fr

ABSTRACT

In wireless mesh networks for unicast traffic, opportunistic network coding are specifically introduced for improved network utilization. Some recent practical implementations like COPE, BEND, DCAR and DODE have shown the promising results over the original 802.11 conventional forwarding mechanism. For better performance, the aim of all mentioned network coding implementations is trying to find more and more coding chances in network topologies. However, they restricted the finding within a simple rule "a pair of coder and decoder". For every coded packet (the combination of natives packets of traffic flows) sent on the traffic flow, we always have one coder (which creates the coded packet) and one decoder (which retrieves the desired packet for the destination). The trivial rule limited coding chances much. In this paper, we are loosing this noose: we decouple the coding and decoding functions from strictly a pair of coder and decoder. For one coder, we can have multiple decoders on the path (up to the number of other traffic flows involved in the coding). With this, more coding chances are found, thus, improving the network performance. We extend our proposed DODE with this new idea, called *Distributed Opportunistic and Diffused Coding with Multiple Decoders* - DODEX. We implement DODEX system in NS-2. The simulation results show that DODEX can outperform its previous introducing systems.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless communication

Keywords

network coding; opportunistic coding; forwarding

1. INTRODUCTION

Since its first introduction in [1], network coding has gained a significant attention from the research communities in the need of improving the way of communication in computer networks. Network coding is a technique which allows the nodes to combine several *native* packets into one *coded* packet for transmission (i.e., coding packets) instead of simply forwarding packets one by one. The network then can save the number of transmissions to reduce the data transfer time and increase the throughput. Based on that, *opportunistic network coding* (ONC) is firstly introduced in COPE [3], the practical network coding system for the unicast traffic. Afterwards, more extended works are presented like BEND [4], DCAR [5] and DODE [6]. Providing a particular coding condition, all these network coding systems can ensure after the coded packets are opportunistically created and broadcasted in the air by a coder (i.e., node intersecting many flows), there will be **one** decoder (i.e., node retrieving the desired packet for the destination) on the traffic flow decoding the coded packet successfully. The trivial rule limited coding chances much. Our previous work DODE has combined the features from COPE (the opportunistic listening), BEND (the diffused gain), and DCAR (the generalized coding condition) to create a better network coding architecture. DODE also exploited the routing metric SPENM (*Shortest Paths of Enriched Neighbors Metric*) which helps to choose the path with the most coding chances. DODE can indeed grasp more coding chance and the obtained throughput is significantly improved.

In this paper, we extend DODE to resolve the restricting rule: we decouple the coding and decoding functions from strictly a pair of a coder and a decoder. After one coder creating the coded packets, multiple decoders can be the next on the transmission path to retrieve the desired packet for the destination. More coding chances can be found, leading to an improved network performance. Our contribution are two-fold:

- We redesign the coding condition in DODEX for each coded packet at the coder, which allows multiple decoders to be on the traffic flow instead of only one in order to find more coding chance and improve network utilization.
- We implement the DODEX system in NS-2 and compare the result with other implementation 802.11, COPE, BEND, DCAR and DODE. The simulation results show that DODEX can outperform these systems.

The remainder of this paper is organized as follows. In section 2, we describe the related work which inspires our proposal. Section 3 details the design of DODEX. Simulation results are presented and analysed in section 4. Finally, we give the conclusion in section 5.

2. RELATED WORK

2.1 Previous Opportunistic Coding systems

By *opportunistically listening* to other transmission, wireless nodes in COPE [3] can use overheard packets to perform ONC. In Fig 1(b), node $0, 2, 3, 4$ wish to send a packet p_0, p_2, p_3, p_4 to its opposite node $2, 0, 4, 3$ respectively, via the forwarder 1. The dashed lines imply related nodes (e.g., 0 and 1) are 1-hop neighbors to each other and the arrows indicate the traffic flows. Node 1 codes four native packets to broadcast ($p_0 \oplus p_2 \oplus p_3 \oplus p_4$) and receivers can extract the desired packets without failure. For examples, node 0 can get p_2 successfully because p_3, p_4 are overheard by transmissions of $(3 \to 1), (4 \to 1))$, and p_0 is its own packet. To perform opportunistic coding, COPE designs its own particular coding condition. Each pair of n native packets will be checked with the coding condition if they can be coded together before nodes can process n-packet coding (the combination of n native packets). By this way, the coders ensure their coded packets are received and decoded successfully. the authors of COPE designed the *two next-hop coding pattern*: two flows intersecting at a node c are codable if the next hop of c on a flow is the previous forwarder of c on the other flow (Fig 1(a)) **or** its neighbor (Fig 1(b)), and vice versa.

BEND argued a problem presented by the nature of ONC: concentrating traffic via a node for coding can lead to packet collisions and drops. BEND suggested the diffused gain, the neighbors of the node intersecting flows (like $3, 4$ of coder I for flows $(0 \to 1 \to 2)$ and $(2 \to 1 \to 0)$ in Fig 1(b)) can actively overhear the transmission and broadcast the coded packets. Instead of only one coder, we can have advantages of *multiple coders* sharing the coding process. From that point of view, BEND keeps the two next-hop coding pattern from COPE, but it increases the number of coders on the flows, which alleviates the packet collisions and drops at the coders, and improves the network performance. DCAR takes another approach for network enhancement: DCAR breaks the two next hop coding pattern provided by COPE. The authors of DCAR creates the general coding condition which allows the coder and decoders not to necessarily be neighbors but any nodes which can intersect flows (coders) and overhear the transmission properly (decoders), respectively. For examples, in Fig 1(c), we can see 3 as coder and 4, 7 as decoders while COPE, and BEND cannot detect that.

3. DISTRIBUTED OPPORTUNISTIC AND DIFFUSED CODING WITH MULTIPLE DECODERS - DODEX

3.1 DODE

In this part, we introduce DODE, which our proposition in this paper is heavily based on. DODE is a network coding architecture that works at MAC layer but also uses information from upper layers (the packet queues and the routing protocol) to do the coding, decoding processes. In summary, DODE contains several features: the opportunistic listening and encoding from COPE, the diffused gain by neighborhood from BEND and the generalized coding condition from DCAR. The coding condition of DODE can be seen as a combination of two features: the diffused gain from BEND and the general coding condition from DCAR. At first, DODE can increase the coding chances by adding more coders (via the diffused gain). Next, the general coding condition is applied, helping DODE to discover the coding chances over the whole traffic paths, better than the two next hop coding pattern from BEND or COPE. Moreover, as already explained in *Related work*, the diffused gain can alleviate the concentrating traffic problem as DCAR inherits it from COPE. Particularly, DODE gains a better performance by taking the advantages form the previous and resolving their problems. Besides, DODE also provide a new routing metric called *SPENM* (*Shortest Path with Enriched Neighbors Metric*) over the routing protocol DSDV [2] to cope with the new coding condition, helping to find routing paths with enhanced coding chances. Due to the lack of space, more technical explanation can be found in our previous work [6].

3.2 Coding chance improvement

As mentioned in *Related work*, each opportunistic coding architecture designs its own coding condition to gain more chances, thus, improving the network performance. In DODEX, coders can combine many native packets for transmission and multiple decoders next on the flow will recover the desired packet. Consider Fig 3(a), we can see three flows $F_1(0 \to 3 \to 4 \to 5)$, $F_2(2 \to 3 \to 6)$ and $F_3(8 \to 7 \to 3 \to 1)$ can be coded together but on the flow F_1 there are two decoders 4 (for leaving out packet p_2) and 5 (removing p_3 to get the desired p_1). With previous implementation, only 2-packet coding can be found instead of 3-packet coding like our proposition DODEX. The coded packets are not maintained *"untouched"* from the coder to the destination but eventually *"peeled out"* the unnecessary packets until only the intended packet can reach to destination. Therefore, the coding condition can be rewritten as followings:

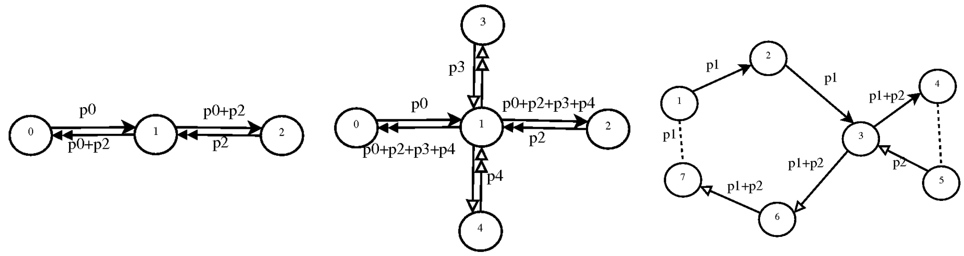
Generalized coding conditions in DODEX
for each pair of n flows F_i and F_j intersecting at node c $c \in ((F_i \cap F_j) \cup (N(nexthop(c, F_i)) \cap N(nexthop(c, F_j)) \cap N(prevhop(c, F_i)) \cap N(prevhop(c, F_j))))$
1. There exists a node $d_{c,i,j} \in (D(c, F_i) \cap (U(c, F_j) \cup N(U(c, F_j)))$. 2. There exists a node $d_{c,j,i} \in (D(c, F_j) \cap (U(c, F_i) \cup N(U(c, F_i)))$.

where:

- F_i denotes the i^{th} flow in the network, $i \in \{1..n\}$
- $nexthop(c, F)/prevhop(c, F)$ denotes the nexthop/previous hop of node c on flow F
- $D(c, F)/U(c, F)$ denotes the downstream/upstream nodes of flow F.
- $N(s_i)$ denotes the set of all neighbors nodes of node s_i
- $d_{c,i,j}$ denotes the decoder of c on flow F_i to remove the packet of flow F_j

3.3 Detailed design

To perform the coding condition above, DODEX has to collect the information *"who sends what"* on the transmission path of each flow and detect the coding chances based on that. Moreover, DODEX needs to store, forward native packets, create and forward coded packets when coding chances exist. Particularly, Each nodes also change their behaviors to also intercept the coded traffic. and have to collect the neighbor list and the source routing list to check



(a) (b) (c)

Figure 1: Opportunistic network coding scenarios

the coding condition above, the queuing system to manipulate packets.

3.3.1 Neighbor list and source routing list

The routing protocol DSDV is modified to cope with carrying more information to detect coding chances. Besides original information of DSDV, each route entry in the update message will also carry the list of nodes constituting the route and the neighbor list of each node along the route. The neighbor list is obtained by collecting the senders' addresses of DSDV messages. Before broadcasting out the DSDV messages, the sender adds its address to the current list of routing path in the entry. After receiving the routing updates from neighbors, each node will process the routing update messages as the original DSDV routing protocol and also cache the information about the neighbors and detailed routing path. With this, all nodes in the network gradually acquire enough information to perform the correct coding.

3.3.2 Queuing system

DODEX uses four different packet queues: Q_{native} for the native packets, Q_ovrhrd for storing the overheard packets, $Q_{codable}$ for the linked lists of packet that can be combined together to create coded packets, and Q_{coded} for coded packets that need forwarding. Figure 2 shows the illustration of queues.

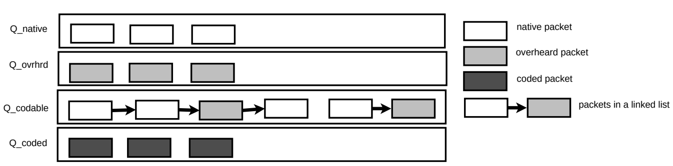


Figure 2: Queuing system

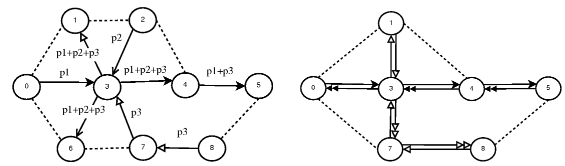
3.3.3 Node behavior

Each node in DODEX has to perform native packet interception, coded packet interception and packet transmission which is described in Section A.

4. SIMULATION AND RESULTS

We use NS-2 as the simulator to compare the performances of DODEX with the previous architectures: IEEE 802.11, COPE, BEND, DCAR and the non-extended DODE. We use two topologies as illustrated in Fig 3(a) and 3(b). The first topology (Fig 3(a)) is provided for test scenario of maximum 3-packet coding with light traffic. The second topology (Fig 3(b)) is used for test scenario of maximum 4-packet coding with stress traffic causing high packet collisions and drops.

Each topology is created in a flat area of $1000m \times 1000m$. The data traffic in the network are all CBR (Constant Bit



(a) Test topology 1 for 3-packet coding (b) Test topology 2 for 4-packet coding

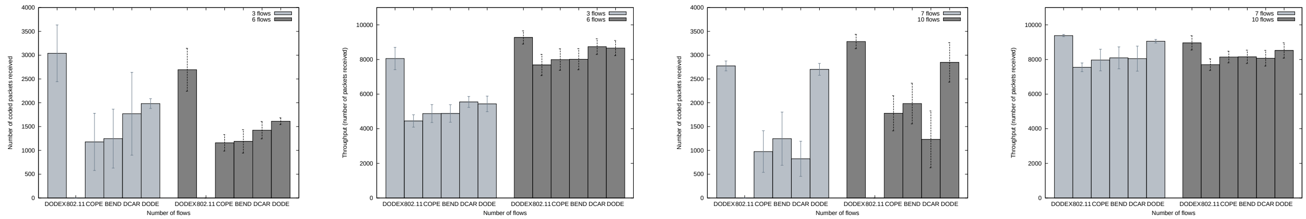
Figure 3: Test scenarios for exposing 3 and 4-packet coding in DODEX

Rate) flows sent over UDP (User Datagram Protocol) using 1000-byte datagrams with an arrival interval of 0.01s and traffic generation duration at source of 150s. The performances is then evaluated by two performance metrics, the throughput and the number of coded packets. We vary the traffic flows in test scenarios as shown in table *Flows in Test scenarios*. Afterwards, the test scenarios will be executed with each traffic-flow variety for all implementations and the result is collected with a 95% confidence interval.

Table 1: Flows in Test scenarios

3 flows varied in test scenario 1	$F_1(0 \to 3 \to 4 \to 5)$, $F_2(2 \to 3 \to 6)$, $F_3(8 \to 7 \to 3 \to 1)$
6 flows varied in test scenario 1	$F_1(0 \to 3 \to 4 \to 5)$, $F_2(2 \to 3 \to 6)$, $F_3(8 \to 7 \to 3 \to 1)$, $F_4(5 \to 4 \to 3 \to 0)$, $F_5(6 \to 3 \to 2)$, $F_6(1 \to 3 \to 7 \to 8)$
7 flows varied in test scenario 2	$F_1(6 \to 2 \to 3 \to 1)$, $F_2(1 \to 3 \to 2 \to 6)$, $F_3(0 \to 3 \to 4 \to 5)$, $F_4(5 \to 4 \to 3 \to 0)$, $F_5(5 \to 6)$, $F_6(1 \to 4)$, $F_7(0 \to 2)$
10 flows varied in test scenario 2	$F_1(6 \to 2 \to 3 \to 1)$, $F_2(1 \to 3 \to 2 \to 6)$, $F_3(0 \to 3 \to 4 \to 5)$, $F_4(5 \to 4 \to 3 \to 0)$, $F_5(5 \to 6)$, $F_6(1 \to 4)$, $F_7(0 \to 2)$, $F_8(5 \to 6 \to 2)$, $F_9(2 \to 6 \to 5)$, $F_{10}(0 \to 2 \to 6)$

As shown in Fig 4(a) and 4(b), with the light traffic, DODEX outperforms previous implementations significantly. In the case of 3 flows, only DODEX discovered the 3-packet coding, and the throughtput gain over the previous is quite promising (32% over our old DODE and 30%-45% over the others). First, COPE and BEND only detects the 2-packet coding between F_2 and F_1. However, coder 3 also serves as a forwarder for F_3, increasing packet collisions and drops at node 3 due to concentrating traffic via the coder. Second, DODE and DCAR are also able to detect the 2 packet-coding between F_1 and F_3 thanks to the general coding condition but the same problem happened too because the diffused gain does not help BEND, DODE or even DODEX much (only 3 as the coder). With 3-packet coding, DODEX allows more packets delivered to destination, result in higher throughput. For the next case (6 flows), because there are



(a) Coded packets result for Test scenario 1 (b) Throughput result for Test scenario 1 (c) Coded packets result for Test scenario 2 (d) Throughput result for Test scenario 2

Figure 4: Test scenario results

more coding chances (F_4, F_5 and F_6 are in reverse directions with F_1, F_2 and F_3, respectively), the previous works can compensate the throughput gain and the number of coded packets by creating the coded packets from these flows with opposite ways. Nevertheless, DODEX still maintains higher throughput because DODEX can detect not only all coding chances like the previous but also the 3-packet coding with multiple decoders.

In the second topology exposing the coding 3 or 4 packets in a single transmission, the performances of all previous and DODEX are presented in Figs 4(c) and 4(d). We would like to check if under heavy traffic, DODEX still keeps the high throughput and coded packets. By applying 4-packet coding with multiple decoders, DODEX transfers more data even in the interference of the non-codable flows. Compared to DODEX, in case of 10 flows (4-6 of them are non-codable), all previous implementations are losing throughput because it takes more transmissions for 2-packet coding. Due to high interference from the non-codable flows, the throughputs gained from all architectures are reduced, both from competition of accessing MAC layer to send coded packets or the packet collisions and drops. DODEX can alleviate the problem by discovering more coding chances via coding with multiple decoders, draining the packets from the queue at the forwarder faster than the others (4-packet coding compared to 2-packet coding), thus, giving DODEX a better performance over the formers (6% over DODE, and 10%-20% over the others as Figs 4(c) and 4(d) shows)

5. CONCLUSION

In this paper, we have proposed a nouvel and practical network coding architecture for unicast traffic in wireless mesh network. We extend the result gained from DODE to explore a new way of intercepting the coding chance in opportunistic network coding, which helps to improve the network performance. We allow many decoders on the flow to share the decoding process instead of only one like the previous systems. More coding chances are discovered, and help to achieve the better network utilization.

6. ACKNOWLEDGMENTS

We appreciate the support and suggestion from Professor Nadia Boukhatem from Computer Science & Network Department, TELECOM ParisTech to accomplish this work.

7. REFERENCES

[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network Information Flow", IEEE Transactions on Information Theory, IT-46, pp. 1204-1216, 2000.

[2] C. E. Perkins, P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers", SIGCOMM 94 -8/94 London England UK, 1994.

[3] S. Katti, H. Rahul, W. H. D. Katabi, M. Medard, J. Crowcroft, "XORs in The Air: Practical Wireless Network Coding", SIGCOMM 06, September 11âĂŞ15, 2006, Pisa, Italy.

[4] J. Zhang, Y. P. Chen, I. Marsic, "MAC-layer proactive mixing for network coding in multi-hop wireless networks", Computer Networks 54 (2010) 196âĂŞ207.

[5] J. Le, J. C.S. Lui, D. M. Chiu, "DCAR: Distributed Coding-Aware Routing in Wireless Networks", IEEE Transaction on Mobile Computing, Vol. 9, No. 4, April 2010.

[6] T. V. Vu, T. M. T. Nguyen, G. Pujolle, "Distributed Opportunistic and Diffused Coding in Multi-hop Wireless Networks", IEEE ICC 2012 Coconet4, June, 2012.

APPENDIX

A. NODE BEHAVIOR

Algorithm 1 Native packet interception

```
p_i ← native packet
if ∃ linked list L ⊂ Q_codable / p_i is codable with ∀p_j, p_j ∈ L then
    Add p_i to the list
else if there exists ∃ p_j ∈ Q_native / p_i is codable with p_j then
    Remove p_i, p_j from Q_native, create the linked list with p_i, p_j and add the list to Q_codable
else if there exists ∃ p_j ∈ Q_ovrhrd / p_i is codable with p_j then
    Remove p_i, p_j from Q_ovhrd, create the linked list with p_i, p_j and add the list to Q_codable
else if p_i is the intended packet then
    Add p_i to Q_native
else
    Add p_i to Q_ovhrd
end if
```

Algorithm 2 Coded packet interception

```
p_i ← coded packet
if p'_i is an intended packet for the node then
    Remove as many as possible the non-intended native packets in p'_i
    if only one desired packet remained then
        Process Algorithm 1 Native packet interception
    else if there exists coded packet p"_i and there still exists decoders on path to destination then
        Add p"_i to Q_coded
    end if
end if
```

Algorithm 3 Packet transmission

```
Access the medium
if Q_codable ≠ ∅ then
    Remove the linked list from Q_codable
    Create the coded packet from the first linked list
    Send the packet
else if Q_coded ≠ ∅ then
    Remove the first coded packet from Q_coded
    Send the packet
else if Q_native ≠ ∅ then
    Remove the first native packet from Q_native
    Send the packet
end if
```