

XCOR: Synergistic Inter-flow Network Coding and Opportunistic Routing

1. Abstract

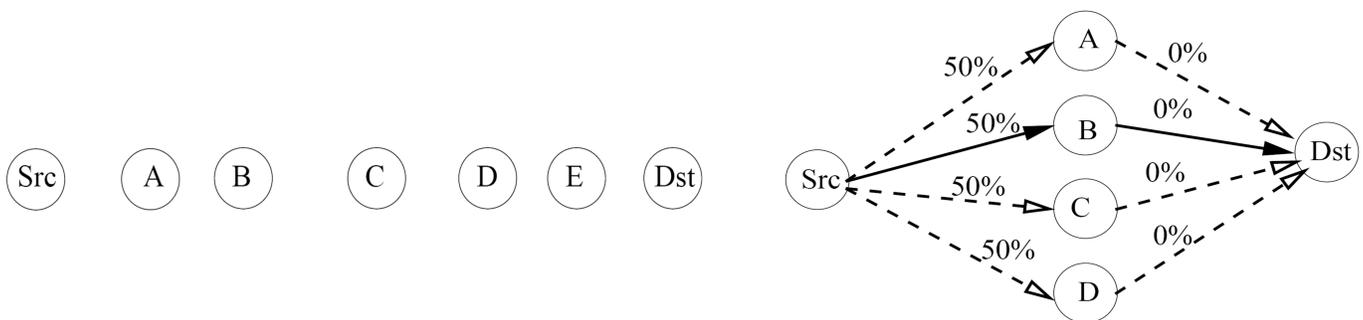
Opportunistic routing (OR) and *inter*-flow network coding (NC) belong to a class of recently proposed, "exotic" optimization techniques that promise several-fold throughput improvement in Wireless Mesh Networks (WMNs). In the past few years, a plethora of new routing protocols have been proposed that exploit either one of these techniques. These protocols are designed to optimize the throughput of either individual flows or jointly for many coexisting flows and have been shown to work well under specific scenarios. This work proposes **XCOR**, the first protocol that integrates interflow (**X**) Network Coding with Opportunistic Routing and exploits their individual potential and their synergy in different scenarios.

2. Problem and Motivation

Wireless mesh networks are characterized by static mesh routers connected by wireless links to each other and to a few gateway nodes. The WMN routers effectively form a multihop wireless access backbone. Recently, the deployment and use of WMNs have increased significantly and several cities have planned and/or deployed WMNs (e.g., [1, 2, 3]). Thus, improving WMN performance will have a direct impact on a growing population of users. Since the most significant application of such networks is to provide broadband Internet access to static or mobile hosts in areas where wired infrastructure is difficult or economically infeasible to deploy, it is important to *optimize the network throughput of WMNs*.

Recent years have witnessed numerous protocols that aim to improve the throughput of WMNs [4, 5, 6, 7, 8, 9]. These protocols are typically built upon, and hence can be decomposed into, one or more building-block techniques that increase throughput in wireless networks. These building blocks include traditional techniques, such as Automatic Repeat Request (ARQ), Forward Error Correction (FEC) or Rate Control (RC), as well as novel, "exotic" techniques, such as opportunistic routing (OR) and *inter*-flow or *intra*-flow network coding (NC), which promise several-fold throughput improvement compared to traditional ones. The common feature of these "exotic" techniques is that they all exploit the broadcast nature of the wireless medium. Protocols that exploit these techniques abandon the traditional 5-layer principle that has guided the design of networking protocols over the past decades and adopt a *cross-layer* approach; they are implemented at a newly defined layer 2.5 and they blend together mechanisms from lower and upper layers.

In contrast to traditional routing which forwards packets along a fixed path from a source to a destination, OR opportunistically exploits multiple paths between the source and the destination. Traditional routing chooses the next hop before transmitting a packet. On the other hand, opportunistic routing sends the packet first and then decides the next hop among all neighbors that hear the packet successfully. Figures 1(a), 1(b) show two typical examples in which opportunistic routing increases throughput compared to traditional routing.



(a) Example 1: Src's transmissions make different amounts of progress towards the destination.

(b) Example 2: Each of Src's transmissions has many independent chances of being received by a node closer to the destination.

Figure 1: Two examples explaining the throughput gain of opportunistic routing over traditional routing.

In Figure 1(a), a traditional routing protocol would select a subset of the intermediate nodes in a given order to form a path from Src to Dst, for example Src-B-D-Dst. In contrast, OR can exploit situations where a transmission from Src is only received by A or is received by A, B, and D. In the first case, A will transmit the packet, thus making some progress, while in the second case D will forward the packet instead of B, thus eliminating one transmission. In Figure 1(b), traditional routing would send all the data packets through the same intermediate node, e.g., through the path Src-B-Dst. In this case, the loss rate of the path is 50%. In contrast, with opportunistic routing *any* of the four intermediate nodes can forward the packet, thus the loss rate drops to only $0.5^4=6.25\%$.

With inter-flow NC, a router can mix (bitwise XOR) packets from different flows and broadcast them as a *single* packet. If the next hop of each flow has already overheard all the mixed packets except for the one destined to it, it can XOR them again with the XORed packet to obtain its own packet. By mixing packets belonging to different flows and transmitting them as one, inter-flow NC reduces the total number of transmissions required, and hence increases the "effective" capacity of the network. In a wireless setting, the basic principle of inter-flow NC can be explained with the simple "Alice-and-Bob" example in Figure 2, from [14]. With NC, the router can code two packets in a single transmission, thus reducing the total number of transmissions required for this exchange from 4 to 3, and increasing the total network throughput by a factor $4/3$.

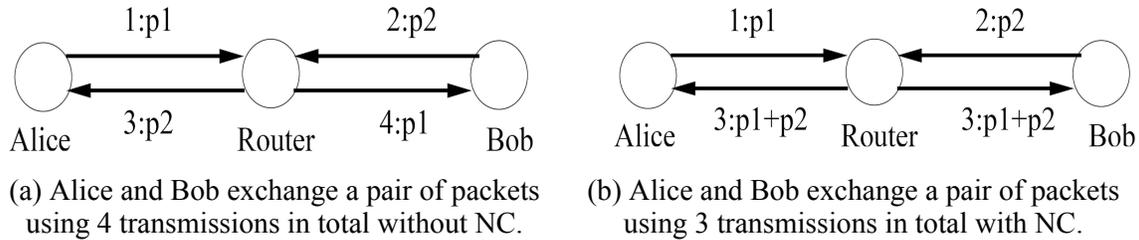


Figure 2: An example showing how NC can improve throughput by increasing the capacity of the network.

While either technique in isolation has been shown to significantly increase the throughput of WMNs, they also pose two intriguing questions: (1) Which of the two techniques performs better and in which scenarios, or, are gains from interflow NC applicable in scenarios where gains from OR are applicable, and vice versa? (2) Can we design a single protocol that exploits both techniques and always outperforms protocols based on either technique alone?

While quantitative analysis of the first question appears quite challenging as conceivably the answer heavily depends on the specific topology and traffic scenarios, qualitative analysis suggests the two techniques can be synergistic if equipped in a single protocol. In general, inter-flow NC is expected to yield poor performance gain in highly lossy environments, since intermediate nodes will not likely have many coding opportunities. In contrast, OR is expected to be more effective in highly lossy environments, because it gives more than one chances for a packet to make some progress towards the destination, and hence can facilitate inter-flow NC by creating more coding opportunities. Conversely, the throughput gain of most OR protocols is reduced as the number of flows in the network increases; while more flows are likely to create more opportunities for inter-flow NC.

In this work, we seek to answer the second question above by designing a new protocol, **XCOR**, that carefully integrates the two techniques. Such an integration poses two major challenges: (1) Special care should be taken, so that the number of duplicate transmissions are avoided (in other words, extra mixing opportunities for NC should not come from duplicate packets). In fact, a preliminary version of COPE [4] discussed the possibility of combining these two techniques and proposed a simple realization. However, the extra gain from adding OR was found to be minimal, and in some cases it even had negative effects exactly due to duplicate transmissions. (2) Interflow NC assumes fixed path routing, i.e., each intermediate router knows the next hops of all the packets it has to forward and decides which ones to mix together. In contrast, in OR, a router does not know in advance the next hop of a packet. Based on this reasoning, [8] claims that combining opportunistic routing with inter-flow NC is **not feasible**.

3. Related work

Several protocols have been recently proposed that exploit either OR (e.g., [4, 5, 6, 7, 8]) or inter-flow NC (e.g., [9, 10]). Recent works have also proposed the idea of *coding-aware* routing [11, 12, 13], which selects routing paths of different flows with the goal of creating more coding opportunities. The majority of these works have assumed *fixed-path* routing. Even in the case of opportunistic routing, we note that the goal of this approach is *different* from ours. In coding-aware routing, routing decisions are always made with the goal of maximizing the benefit from

NC, even in scenarios which are inherently unfavorable to NC. In contrast, our work seeks to synergistically combine the two techniques, in order to exploit the benefits of *both* or *either of them*, under different scenarios.

Several protocols have been proposed that integrate OR with *intra-flow* NC (e.g., [4, 5, 6]). This approach is orthogonal to ours. With *intra-flow* NC, routers mix packets belonging to the same flow and transmit them as a single packet, with the goal to remove the need for coordination in OR. Removing the need for coordination can improve throughput of individual flows. However, protocols that combine OR with *intra-flow* NC still suffer from other problems inherent to OR, e.g., performance gain drops as the number of flows increases. In contrast, *inter-flow* NC mixes packets from different flows with the goal to increase the network capacity, which results in throughput improvements for each individual flow, and the gain becomes more prominent under heavy congestion (i.e., in the presence of many flows). Actually, a protocol that would integrate OR with both types on NC can be seen as the ultimate challenge, however, practical challenges make such a protocol very hard to realize in practice.

4. Approach and Uniqueness

The design of **XCOR** comprises two main components: (1) Opportunistic Routing, (2) Packet encoding/decoding algorithm.

Opportunistic Routing. The OR component of our protocol is inspired by the SOAR protocol [10]. In contrast to other OR protocols (e.g., ExOR [2], ROMER [11], MORE [3]), SOAR was designed to facilitate multiple flows, which makes it easier to integrate with *inter-flow* NC. The primary problem in the design of any OR protocol is the selection of the *forwarding list*, i.e., the set of the mesh routers that are responsible for forwarding packets to the destination. In **XCOR**, the forwarding list for each flow forms a *thin belt* along the default (shortest ETX [14]) path from the source to the destination. In contrast to previous protocols (e.g., [4], [8]) that aggressively add forwarders to the list, the thin-belt approach, first introduced in SOAR, adds only forwarders close to the default routing path. This ensures that with high probability each forwarding node overhears the transmissions of neighboring nodes, and can avoid duplicate transmissions. Also, to further limit the total number of forwarding nodes and reduce the probability of congestion, it limits the maximum number of possible next hops at each router to a small number h_{max} .

Similar to every other OR protocol, in **XCOR** packets are broadcast at each hop. After each transmission, the node that is closest in ETX to the destination, among the possible next hops, forwards the packet immediately, while the other nodes set their forwarding timers according to their priority (distance to the destination). If they overhear the transmission from a node of higher priority, they cancel their timers. Hence, after each transmission, with high probability (thanks to the thin-belt approach) only one node forwards the packet, with priority given to the nodes that make larger progress towards the destination.

Packet Encoding/Decoding. **XCOR**'s encoding/decoding algorithm seamlessly integrates *inter-flow* NC as in COPE [7] with opportunistic routing. As in COPE, to limit packet reordering and to reduce coding delays, we use per-flow virtual queues and consider only the packets at the heads of the virtual queues when we look for coding opportunities. This also reduces the complexity of the searching algorithm. Finally, as in COPE, we only perform opportunistic coding and never delay packets at the routers, waiting for coding opportunities to arise.

In COPE, packets are selected for encoding with the objective that with high probability each of the next hops will be able to decode the coded packet and obtain its native packet. In **XCOR**, each packet has more than one possible next hops. When a node transmits a packet, OR gives priority to the next hop closest to the destination and furthest from the previous hop, in order to maximize the progress made to the destination. However, the next hop furthest from the current transmitter has in general a smaller probability to decode a coded packet that contains a mixture of packets belonging to flows crossing the current node, since the probability that it has overheard some of those packets is lower. Hence, there is a **tradeoff** between the next hop selection made by OR and the selection of packets to be encoded.

Assume a node R has M different flows that use it as one of their hops towards their destinations. Let F_i , $i = 1, 2, \dots, M$, denote the i -th flow, D_i denote the destination of flow F_i , and p_i is the packet at the head of the virtual queue of flow F_i . For each flow F_i , node R has a set of next hops N_i and let $N_{ij} \in N_i$, $j = 1, 2, \dots, h_i$, denote the j -th next hop for flow F_i at node R . With any set of N packets (i.e., from N different flows) encoded together, we calculate a utility gain U as the sum of the gains for each of the different flows whose packets are mixed in the encoded packet.

$$U = \sum_{i=1}^N U_{R_i} \quad (1)$$

The gain from NC is reflected in the summation of the gains from OR for each individual flow. For the i -th flow F_i , there are h_i possible next hops N_{ij} , $j = 1, 2, \dots, h_i$, each of them with an associated gain UR_{ij} . Hence

$$U_{R_i} = \sum_{j=1}^{h_i} U_{R_{ij}} \quad (2)$$

By selecting N_{ij} as the next hop for flow F_i , we achieve an improvement in terms of ETX towards the destination D_i , equal to $ETX(R, D_i) - ETX(N_{ij}, D_i)$, where $ETX(X, Y)$ is the length of the shortest path (in terms of ETX) from node X to node Y . For this progress to be made, the packet has to be received by node N_{ij} , when it is broadcast by R ; this happens with probability

$$P_R(N_{ij}) = 1/ETX(R, N_{ij}) \quad (3)$$

Also, in case the packet is a mixture of N native packets, node N_{ij} needs to be able to decode it and obtain the native packet destined to it, p_i , otherwise no progress is made for flow F_i . Let $P_D(N_{ij})$ be the probability that node N_{ij} is able to decode the packet and obtain p_i , then according to [7]

$$P_D(N_{ij}) = \prod_{k=1, k \neq i}^N P\{N_{ij} \text{ has } p_k\} \quad (4)$$

Similar to [7], $P\{N_{ij} \text{ has } p_k\} = 1$, if N_{ij} had previously announced packet p_k or it had transmitted it to R , and $P\{N_{ij} \text{ has } p_k\} = 1/ETX(\text{prevhop}(R), N_{ij})$, in any other case, with $\text{prevhop}(R)$ being the node that transmitted p_k to R . Summarizing, we finally get

$$\begin{aligned} U_{R_{ij}} &= P_R(N_{ij}) \cdot P_D(N_{ij}) \cdot [ETX(R, D_i) - ETX(N_{ij}, D_i)] \\ &= \frac{1}{ETX(R, N_{ij})} \cdot \left(\prod_{k=1, k \neq i}^N P\{N_{ij} \text{ has } p_k\} \right) \cdot \\ &\quad [ETX(R, D_i) - ETX(N_{ij}, D_i)] \end{aligned}$$

Then (1) gives:

$$\begin{aligned} U &= \sum_{i=1}^N \sum_{j=1}^{h_i} \frac{1}{ETX(R, N_{ij})} \cdot \\ &\quad \left(\prod_{k=1, k \neq i}^N P\{N_{ij} \text{ has } p_k\} \right) \cdot [ETX(R, D_i) - ETX(N_{ij}, D_i)] \end{aligned} \quad (5)$$

If node R has packets belonging to M different flows to forward, we need to consider $M - 1$ different packet combinations in order to find the one that gives the largest utility. To reduce the time complexity of the algorithm, we apply a simple heuristic. Our algorithm starts by dequeuing the packet at the head of the FIFO output queue (i.e., the first among the packets whose forwarding timer has expired), and then it goes over the remaining $M - 1$ flows considering only the packet at the head of each virtual queue (i.e., we consider a total of only M packets). For each packet, it decides to mix it if the utility obtained after mixing this packet is larger than the utility obtained before considering it. Different from COPE, we do not examine the remaining flows in a random order but in a decreasing order of their virtual queue lengths, i.e., we give priority to heavily loaded flows and try to minimize packet dropping for all flows. Finally, as in COPE, we only perform opportunistic coding and never delay packets at the routers.

5. Results and Contributions

We evaluated the performance of **XCOR** using the *Qualnet* simulator [10], a widely used wireless network simulator with a detailed signal propagation model. Our experiments focus on a few concrete scenarios to achieve the following two objectives. First, we compare XCOR against previous state-of-the-art protocols SOAR (OR protocol), COPE (interflow NC protocol) and Srcr (traditional routing) to show the added benefits from combining the two techniques. Second, we show that the relative performance benefit of individual building-block techniques (interflow NC or OR) very much depends on the specific topology and traffic scenario.

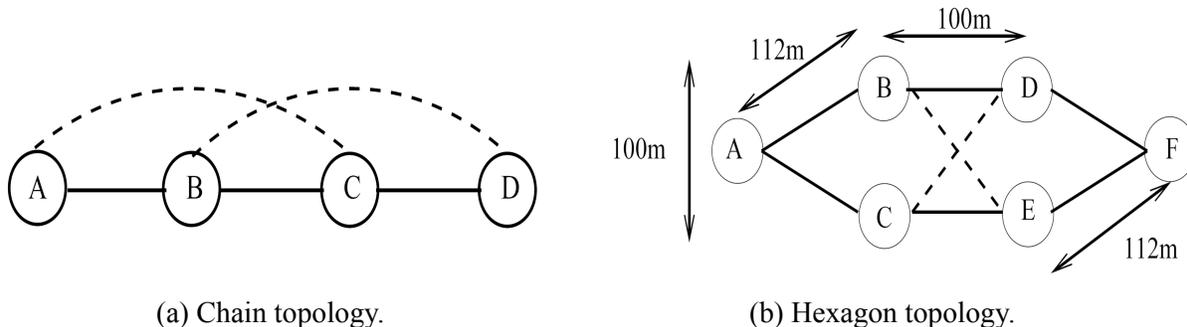


Figure 3: Two topologies used in simulation evaluation. Solid lines represent high-quality links, and dashed lines represent low-quality links.

We used two different topologies shown in Figures 3(a), (b), reflecting two typical examples in which OR increases throughput compared to traditional routing. In each case, we generate two UDP flows (A→D and D→A in Figure 5(a) and A→F and F→A in Figure 5(b)). The results for the two topologies are shown in Figures 4(a), (b), respectively.

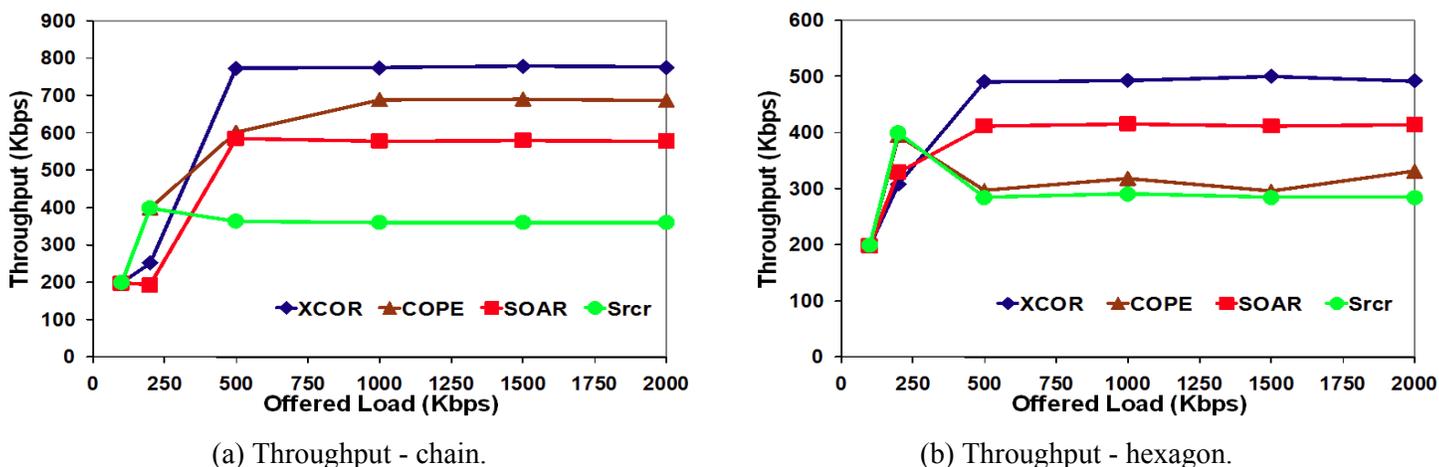


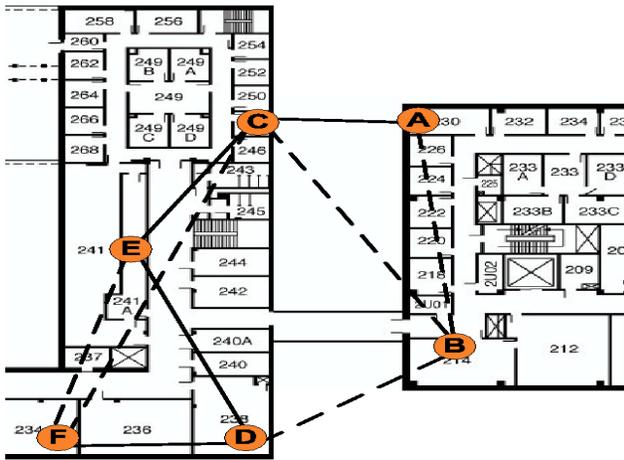
Figure 4: Throughput comparison of Srcr, SOAR, COPE, and XCOR in two different topologies.

We observe that for medium and high loads, **XCOR** outperforms **Srcr**, **SOAR**, and **COPE**, by 115%, 34%, and 13%, respectively, in the chain topology, and by 76%, 22%, and 70%, respectively, in the hexagon topology. The different gain over **COPE** and **SOAR** are due to the different relative performance of **COPE** and **SOAR** in the two scenarios. In the chain topology, **COPE** outperforms **SOAR** by 19%. This scenario favors inter-flow NC over OR because the one-hop links used by **COPE** are of good quality, offering many opportunities for packet mixing, while the two-hop links that are opportunistically used by OR are of medium to low quality. On the other hand, in the hexagon topology, **COPE**'s throughput is only slightly higher than **Srcr**'s and 28% lower compared to **SOAR**. The high losses in this scenario prevent **COPE** from finding many opportunities for mixing, and hence the reduction in the number of transmissions from NC is small. In addition, in some cases, **Srcr** (also used in **COPE**) selects the upper path for one flow and the lower path for the other one, offering no coding opportunities at all. In contrast, OR offers most of the time two available next hops at each node. This explains the large gain of OR over traditional routing. In addition, it creates more opportunities for packet mixing, which translates into additional throughput gain of **XCOR** over **SOAR**.

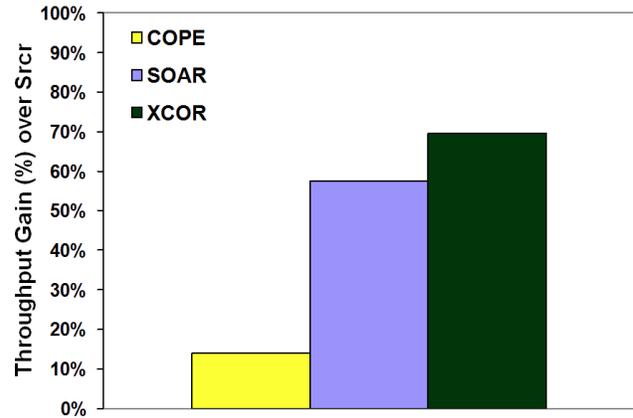
In summary, results on these two topologies show that there is no clear answer to the question which of the two building block techniques (OR or inter-flow NC) offers higher throughput improvement; the final answer depends on the specific network topology and traffic. However, equipped with both two techniques, **XCOR** can exploit their individual potential and their synergy in any scenario.

We have also conducted a preliminary evaluation of **XCOR** on an 802.11 WMN testbed consisting of 6 nodes placed in an office building. Each node has an Atheros 5212-based 802.11a/b/g wireless radio operating in b mode. Each mesh router runs Mandrake Linux 10.1 (kernel 2.6.8-12) and the open-source madwifi driver [11] is used to enable the wireless cards. We used the publicly available implementation of **COPE** from the authors of [7], which also offers an implementation of the **Srcr** protocol. **COPE** is implemented in Linux using the Click toolkit [12]. We leveraged this implementation in implementing **SOAR** and **XCOR**. Figure 5(a) shows a schematic of the testbed and Figure 5(b) shows the

results which confirm our simulation findings, showing again that **XCOR** outperforms both SOAR and COPE.



(a) Testbed Topology.



(b) Testbed evaluation results.

Figure 5: Testbed evaluation.

Significant contributions of our work are:

- Our work is the first to study the integration of OR and inter-flow NC, two novel techniques that significantly increase throughput of WMNs, each under different scenarios.
- We identify the challenges faced by such an integration, which made researchers in past to suggest that such an integration is infeasible; We also identify the synergy between the two techniques that actually makes such an integration feasible.
- We design the **XCOR** protocol that carefully integrates the two techniques.
- Our simulation evaluation shows that there is no clear answer to the question which of the two building block techniques (OR or inter-flow NC) offers higher throughput improvement; the final answer depends on the specific network topology and traffic. However, equipped with both techniques, **XCOR** can exploit their individual potential and their synergy in any scenario, thus advancing the state-of-the-art in the design of high-throughput routing protocol design in WMNs.
- Following the recent trend in the WMN community of demonstrating the performance benefits of a new protocol on a testbed, we also implemented a preliminary version of **XCOR** on commodity 802.11 hardware. Our preliminary evaluation on a 6-node WMN testbed confirms the simulation results and shows that **XCOR** can indeed be implemented in practice.

In the future, we plan to conduct an extensive evaluation of **XCOR** in large networks, under different topologies, and different traffic patterns. A fundamental question that has to be answered before conducting such a detailed evaluation is *what is the most appropriate methodology* for the evaluation of protocols based on "exotic" optimization techniques. Our recent work in [13] identifies the challenges posed in the evaluation of this new class of protocols, discusses the diverse set of current practices in evaluating recently proposed protocols and their strengths and weaknesses, showing that there is an urgent need to carefully rethink the implications of the new cross-layer routing protocol design on the evaluation methodology. Finally, it makes several concrete suggestions on the desired evaluation methodology for meaningful and fair comparison of such protocols.

5. Acknowledgment

This work was supported in part by NSF grant CNS- 0626703.

6. References

- [1] "MIT Roofnet," <http://www.pdos.lcs.mit.edu/roofnet>.
- [2] Technology For All (TFA). <http://tfa.rice.edu>.
- [3] Google WiFi. <http://wifi.google.com/>.
- [4] S. Biswas and R. Morris. ExOR: Opportunistic multi-hop routing for wireless networks. In Proc. of ACM SIGCOMM, 2005.
- [5] Y. Yuan, H. Yang, S. H. Wong, S. Lu, and W. Arbaugh. ROMER: Resilient opportunistic mesh routing for wireless mesh networks. In Proc. of IEEE WiMesh, 2005.
- [6] E. Rozner, J. Seshadri, Y. Mehta, and L. Qiu. Simple opportunistic routing protocol for wireless mesh networks. In Proc. of IEEE WiMesh, 2006.
- [7] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft. XORs in the air: Practical wireless network coding. In Proc. of ACM SIGCOMM, 2006.
- [8] S. Chachulski, M. Jennings, S. Katti, and D. Katabi. Trading structure for randomness in wireless opportunistic routing. In Proc. of ACM SIGCOMM, 2007.
- [9] C. Gkantsidis, W. Hu, P. Key, B. Radunovic, S. Gheorghiu, and P. Rodriguez. Multipath code casting for wireless mesh networks. In Proc. of ACM CoNEXT, 2007.
- [10] QualNet. <http://www.scalable-networks.com>.
- [11] madwifi. <http://madwifi.org>.
- [12] R. Morris, E. Kohler, J. Jannotti, and M. F. Kaashoek. The click modular router. In Proc. of ACM SOSP, 1999.
- [13] D. Koutsonikolas, Y. C. Hu, and K. Papagiannaki. How to evaluate exotic wireless routing protocols? In Proc. of ACM HotNets-VII, 2008.
- [14] D. S. J. De Couto, D. Aguayo, J. C. Bicket, and R. Morris. A high-throughput path metric for multi-hop wireless routing. In Proc. of ACM MobiCom, 2003.