XCOR: Synergistic Interflow Network Coding and Opportunistic Routing

Dimitrios Koutsonikolas, Y. Charlie Hu and Chih-Chun Wang
Purdue University, West Lafayette, IN, USA
{dkoutson, ychu, chihw}@purdue.edu

1. MOTIVATION

In the past few years, a plethora of new routing protocols have been proposed that improve the throughput of wireless mesh networks (WMNs) [2, 11, 10, 7, 5, 3]. Two of the building blocks shown to achieve significant performance benefits are opportunistic routing (OR) and interflow network coding (NC), both exploiting the broadcast nature of the wireless medium.

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. OR broadcasts the packet first and then decides the next hop among all neighbors that hear the packet successfully, thus providing more chances for a packet to make some progress towards the destination. Interflow network coding exploits the broadcast nature of the wireless medium by reducing the number of transmissions needed for forwarding packets belonging to different flows, and hence increases the "effective" 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?[1] (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, interflow 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 interflow 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 interflow coding.

[1]This interesting question was also posted by Baskaran Raman at SIGCOMM 2006 online discussion site.

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 [6] 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 interflow NC is not feasible.[2]

In this following, we first present the design of XCOR that integrates interflow NC (**XC**) with **O**pportunistic **R**outing. We then present a preliminary evaluation of XCOR which substantiates our claim that the answer to our posed question (1) above very much depends on the specific topologies and traffic scenarios.

2. XCOR DESIGN

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 interflow NC.

The forwarding list selection forms a *thin belt* along the default (shortest ETX) path. In SOAR, the source includes only the default path and its own possible next hops in the packet header, and every forwarder selects and updates the header with its own next hops close to that path. This means that the routing header of a packet is updated at each hop, as the packet travels towards the destination. To simplify encoding/decoding operations, we want the routing header of a packet to remain unchanged as the packet travels in the

[2]We note here that OR has been combined with *intra-flow* NC [3, 5], where NC is used as a technique for reducing the coordination overhead required by OR protocols.

network. For this reason, in XCOR the source computes recursively the candidate next hops for each possible next hop towards the destination, and stores them in the header, sorted in terms of ETX-based proximity to the destination.

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 scheme seamlessly integrates inter-flow NC (as in COPE) with OR. 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 $F_i,\ i = 1, 2, ...M$, 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 $\mathbb{N}_i$ and let $N_{ij} \in \mathbb{N}_i,\ j = 1, 2, ..., 5$, denote the j-th next hop for flow F_i at node R.

With any set of N packets (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.

$$U = \sum_{i=1}^{N} U_{R_i} \tag{1}$$

The gain from NC is reflected in the summation of the gains from OR for each individual flows. For the i-th flow F_i, there are $h_i \leq 5$ possible next hops $N_{ij},\ j = 1, 2, ..h_i$, each of them with an associated gain $U_{R_{ij}}$. Hence

$$U_{R_i} = \sum_{j=1}^{h_i} U_{R_{ij}} \tag{2}$$

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

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

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]

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

Similar to [7], $P\{N_{ij}\ has\ p_k\} = 1$, if N_{ij} had previously announced packet p_k or it had transmitted it to R. In any other case, node R uses the ETX metric to estimate (guess) $P\{N_{ij}\ has\ p_k\} = 1/ETX(prevhop(R), N_{ij})$, with $prevhop(R)$ being the node that transmitted p_k to R.

Summarizing, we finally get

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

Then (1) gives:

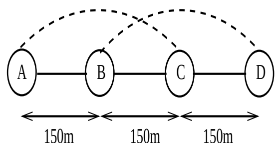
$$\begin{aligned} U = \sum_{i=1}^{N}\sum_{j=1}^{h_i} \frac{1}{ETX(R, N_{ij})} \cdot \\ (\prod_{k=1, k\neq i}^{N} P\{N_{ij}\ has\ p_k\}) \cdot [ETX(R, D_i) - ETX(N_{ij}, D_i)] \end{aligned} \tag{5}$$

If node R has packets belonging to M different flows to forward, we need to consider $2^M - 1$ 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.

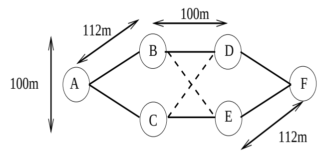
Feedback for Local Recovery and Packet Mixing.

XCOR uses reception reports to provide feedback for local recovery[3] and packet mixing. Reception reports include information about the last packets a node has heard. Every node processes all the reception reports it receives from its

[3]XCOR offer only *best effort* reliability.

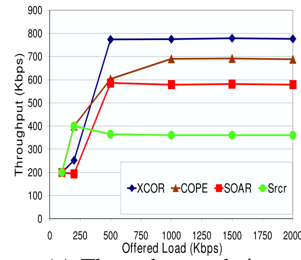


(a) Chain topology.

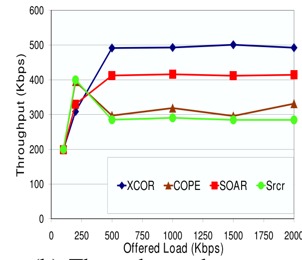


(b) Hexagon topology.

Figure 1: Two topologies used in simulation evaluation.



(a) Throughput - chain.



(b) Throughput - hexagon.

Figure 2: Throughput comparison of Srcr, SOAR, COPE, and XCOR in two different topologies.

neighbors and uses them to update either its retransmission events (if it had transmitted in the past the reported packet) or its view of what packets the sender of the reception report has, which will be used by the packet encoding algorithm. XCOR piggybacks reception reports on data packets every time it has a chance, and it uses periodic control packets when it does not have enough data packets to send. To reduce the feedback overhead, reception reports in XCOR are cumulative, in the form of a bitmap that includes information about the last K packets of a flow.

3. EVALUATION

We evaluate the performance of XCOR using the Qualnet simulator [9]. Our experiments focus on a few concrete scenarios to achieve the following two objectives. First, we compare XCOR against previous protocols SOAR (opportunistic routing), COPE (interflow NC) 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.

We used the 802.11b MAC protocol and a nominal rate of 2 Mbps, and disabled autorate adaptation and RTC/CTS. We used the two-ray propagation model and added random noise to make the setting more realistic. The results are averaged over 10 different seeds.

We used two different topologies shown in Figures 1(a) and Figure 1(b), reflecting two typical examples in which OR increases throughput compared to traditional routing. In each case, we generate two UDP flows ($A \rightarrow D$ and $D \rightarrow A$ in Figure 1(a), $A \rightarrow F$ and $F \rightarrow A$ in Figure 1(b)). The chain topology allows OR to exploit "long jumps" (e.g., from A directly to C), while the hexagon topology offers two different paths connecting each source-destination pair. In both figures solid lines represent high-quality links, and dashed lines represent low-quality links.

We observe that for medium and high loads, XCOR outperforms the 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 interflow NC) offers higher throughout 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.

In the future, we plan to to conduct detailed performance evaluation of XCOR under many different scenarios and traffic patterns. We are also developing an implementation of XCOR on a WMN testbed.

4. REFERENCES

[1] J. Bicket, D. Aguayo, S. Biswas, and R. Morris. Architecture and evaluation of an unplanned 802.11b mesh network. In *Proc. of ACM MobiCom*, 2005.

[2] S. Biswas and R. Morris. ExOR: Opportunistic multi-hop routing for wireless networks. In *Proc of ACM SIGCOMM*, 2005.

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

[4] R. Draves, J. Padhye, and B. Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *Proc. of ACM MobiCom*, 2004.

[5] 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.

[6] S. Katti, D. Katabi, W. Hu, H. Rahul, and M. Medard. The importance of being opportunistic: Practical network coding for wireless environments. In *Proc. of Allerton Conference*, 2005.

[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*, August 2006.

[8] C. Qin, Y. Xian, C. Gray, N. Santhapuri, and S. Nelakuditi. I^2MIX: Integration of Intra-flow and Inter-flow Wireless Network Coding. In *Proc. of IEEE WiNC*, 2008.

[9] QualNet. http://www.scalable-networks.com.

[10] E. Rozner, J. Seshadri, Y. Mehta, and L. Qiu. Simple opportunistic routing protocol for wireless mesh networks. In *Proc. of IEEE WiMesh*, 2006.

[11] 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.